[exibição de vídeo]

NARRAÇÃO: *Já imaginou se você não tivesse seu direito à liberdade de expressão respeitado? Se seus dados particulares fossem divulgados sem autorização? Ou se você tivesse a sua navegação filtrada por causa de interesses comerciais? A liberdade de expressão, a privacidade e a neutralidade da rede são fundamentais para a internet. E esses são apenas alguns dos dez princípios formulados pelo Comitê Gestor da Internet no Brasil. O CGI.br promove há mais de 20 anos a internet no país. Graças a ele, você tem constante inovação, segurança e diretrizes para o desenvolvimento da internet. Isso tudo é feito de forma colaborativa transparente e democrática.*

O comitê é formado por representantes de todos os setores da sociedade. Assim, as decisões são tomadas por meio do diálogo, com a participação de todos os membros, até que um consenso seja alcançado. Por esses e outros fatores, o modelo brasileiro de governança se tornou referência no mundo todo. CGI.br, por uma internet cada vez melhor no Brasil.

NARRAÇÃO: *Quando você registra um domínio .br, você está contribuindo para a melhoria da internet no Brasil. Pois cada domínio que usa o .br é registrado pelo NIC.br, o Núcleo de Informação e Coordenação do Ponto BR, que, além de registros de nomes de domínios, investe em análise e tratamento de incidentes de segurança, projetos de tecnologias de redes e operações, pesquisas que trazem indicadores sobre o uso das tecnologias da informação e comunicação, implementação de pontos de troca de tráfego local na internet, projetos que contribuem no desenvolvimento global da web e muito mais.*

Tudo isso porque o NIC.br é uma entidade civil de direito privado e sem fins lucrativos, que mantém uma estrutura de registro de domínio segura, estável e de confiança. E reverte parte do que você paga pelo seu domínio no desenvolvimento de infraestrutura, trazendo benefícios para todos que usam a internet no Brasil. Toda essa inovação, tecnologia de ponta, segurança e infraestrutura só é possível porque você tem um domínio .br. NIC.br, sempre em busca do melhor para a nossa internet.

SR. ADRIANO CANSIAN: Olá, bom dia. Sejam todos bem-vindos ao GTS 35, com um evento do Comitê Gestor da Internet no Brasil, organizado pelo Núcleo de Informação e Coordenação do Ponto BR, NIC.br e o Registro.br.

Nessa manhã gostaríamos aí, de agradecer a todos que estão conosco nessa transmissão, nesse evento diferente do GTS. Nós, que estávamos sempre acostumados a nos reunirmos nessa época, em São Paulo, todos os anos, dentro da Semana de Infraestrutura da Internet no Brasil, esse ano estamos fazendo, excepcionalmente, esse evento online com os votos e o nosso desejo que nós possamos estar todos

juntos o ano que vem, de novo, para nos reunirmos pessoalmente. Então, essa versão diferente do GTS 35 nessa manhã, novamente eu dou as boas-vindas a todos.

E agradeço principalmente aos nossos apresentadores que vão compartilhar o seu conhecimento e as suas informações relevantes na área de segurança da informação, de segurança de Internet, segurança de rede, aqui conosco hoje.

Então, imediatamente eu gostaria de convidar o Rafael Obelheiro, da Udesc, o Prof. Dr. Rafael Obelheiro, da Udesc, e o Tiago Heinrich, da Universidade Federal do Paraná, que vão falar para a gente sobre ataques DDoS por reflexão: visão de um honeypot multiprotocolo. Rafael e Tiago, a apresentação será feita pelo Tiago, sejam muito bem-vindos. A palavra é de vocês.

SR. TIAGO HEINRICH: Então, primeiramente, bom dia. Por enquanto, eu... Então, primeiramente bom dia. Eu sou o Tiago, eu estarei apresentando a nossa pesquisa, que vem sendo desenvolvida por alguns anos já, que é Ataques DDoS por reflexão: visão de um honeypot multiprotocolo. Eu sou o Tiago, essa pesquisa é minha e do Prof. Rafael.

Se nós acabarmos observando o tópico, então, o ponto de DDoS em específico, é possível ver um fator histórico, então, é sempre legal discutir um pouco, que não é uma coisa nova e está presente durante anos, esses ataques na Internet. Então, um ponto que teve um conjunto de ataques... um ataque específico que teve muita relevância nesses últimos 20 anos. E ele só depende do spoofing de um endereço. Então, se eu tenho a possibilidade de fazer o spoofing, eu vou conseguir estar realizando esse ataque. E eu tenho vantagens como realizar amplificações, que vão acabar auxiliando muito os atacantes específicos.

Eu acabei listando um conjunto de notícias dos últimos anos, que é o período que a gente acabou desenvolvendo essa pesquisa. Então, na mídia, é bem prático e é um assunto que eles acabam abordando bastante, o volume que esses ataques acabam atingindo, como o número de organizações que acabam vendendo esses serviços e organizações também, que foram atacadas. Se pegar a cada ano, existe alguma organização maior que acabou sendo atacada por ataques DDoS.

Com esse intuito, a ideia dessa apresentação é basicamente passar uma breve contextualização da atualidade, então o que está se passando, eu vou trazer um pouco de estatísticas que acabam sendo realizadas a cada semestre, sobre os ataques específico. Vou discutir sobre tendências, então, duas tendências de DRDoS específicas, que foram o que nós tivemos enfoque ao decorrer desses anos de estudo. Vou apresentar o nosso honeypot, que é o nosso recurso principal para

coleta de dados, então como a gente faz essas coletas, que tipo de... qual a posição nossa na rede, que definições a gente está considerando para essas observações. E, por fim, eu vou apresentar resultados mais práticos dos volumes de tráfego e que tipo de conteúdo esse honeypot em si está capturando.

Observando, então, o contexto atual, da atualidade, peguei algum conjunto de estatísticas da Netscout. Existem inúmeras estatísticas na Internet, a maioria é similar. Então, se alguém quiser dar uma olhada na Netscout, Akamai, existem inúmeras estatísticas todos os semestres, que acabam abordando o crescimento desses ataques na Internet. Então, considerando o nosso período que começou essa coleta, em 2018, o primeiro ponto que é legal abordar, é que de 2017 para 2018, houve um crescimento de 9% de ataques DRDoS. Então, os ataques que acabam utilizando refletores em específico. Se considerar essa mesma avaliação de 2019, 2020, nós temos mais um crescimento de 15%. Então, é um ataque que, ao longo dos anos, nunca tende a diminuir, ele sempre tende a crescer a sua frequência no ambiente.

Considerando só o período do lockdown, então, o Covid, que acabou levando grande parte, a maioria da população a ficar em isolamento, nós tivemos, só nesse período, um crescimento de 25%. E se nós considerarmos em 2019, ataques DDoS em geral, então, genericamente todos os ataques DDoS, 70% deles acabam utilizando refletor. Que é o ponto que a gente vai acabar abordando mais nessa apresentação.

Então, o que seria um ataque DRDoS? Distributed Reflection Denial of Service. Ele tem esse R no meio porque estão explorando refletores. Qual que é a vantagem desses refletores? Eu acabo colocando mais um nível, mais uma camada entre o atacante e as vítimas. Eu consigo gerar conjunto maior ainda de tráfego e eu consigo explorar múltiplos refletores ao mesmo tempo. Eu não estou limitado à utilização de somente um refletor, que é o ponto que a gente vai acabar discutindo um pouco. Claro que a utilização desses refletores traz mais vantagens ainda para o atacante, porque ele não tem a necessidade de utilizar também, um número maior de bots, que era o que acontecia há alguns anos atrás.

Então, a primeira tendência que a gente vai acabar abordando são ataques multiprotocolos. Então, é um ataque DRDoS, então ele está utilizando um refletor. Então, o bot não está enviando o tráfego diretamente para a vítima, o bot, ele está explorando um recurso intermediário e esse recurso intermediário vai ser responsável por enviar esse conjunto de pacotes para a vítima. Então, esse elemento no meio aqui, é um refletor. Existem inúmeros refletores que os atacantes podem utilizar, geralmente são protocolos Legacy e

aplicações mal configuradas, que eu vou apresentar um conjunto que a gente acabou utilizando para esse estudo. Essa camada a mais, é uma camada ainda mais que acaba dificultando a identificação do atacante, e ela também é responsável por realizar a amplificação desse tráfego. Então, o bot, ele pode enviar uma requisição que quando chegar nesse refletor, ele vai gerar um conteúdo ainda maior, e esse conteúdo será encaminhado para a vítima.

Outro ponto que a gente vai também, acabar discutindo são os ataques carpet bombing. Esses caras, eles são bem mais recentes, que eles também vão explorar um refletor. Eles têm um outro ponto que acaba dificultando a mitigação e a detecção, que é a distribuição do tráfego entre um bloco CIDR, então entre um conjunto de vítimas. Eu não vou estar somente atacando um único host, então, um único endereço, eu estou atacando um bloco inteiro. A vantagem desse ataque é que eu vou saturar a rede em si, e eu vou estar conseguindo distribuir o conjunto de pacotes que iria para uma única vítima, entre um conjunto de vítimas.

Quando a gente chegar mais para os resultados reais que a gente observou, vai ser possível identificar porque esse ataque acaba dificultando ainda mais a mitigação.

Então, a única diferença de um ataque carpet bombing para um ataque DRDoS, é que eu vou ter essa distribuição de pacotes entre um conjunto de vítimas. Eu não estou atacando uma única vítima. Eu vou ter uma camada a mais, que vai dificultar a identificação, como o DRDoS, mas eu vou estar distribuindo esse conjunto de tráfego entre um conjunto de vítimas.

Agora, para o conjunto de dados que a gente acabou coletando, essa coleta foi permitida por causa de um honeypot, esse honeypot tem a função de um refletor, então, ele é um elemento que vai se passar por um refletor na rede. Então, os atacantes vão identificar esse indivíduo, vão começar a realizar requisições, vão ver que as requisições são válidas e vão explorar esse recurso para ataques de amplificação, de amplificação e de reflexão. Esse honeypot, nós demos o nome dele de MP, MPH, então, multiprotocol honeypot, porque ele não está limitado a somente um protocolo. Muitos trabalhos na literatura acabam só considerando um único protocolo, esse honeypot, ele considera nove protocolos. Então, eu tenho nove protocolos implementados dentro desse honeypot: Chargen, CoAP, CLDAP, NTP, QOTD, SSDP e Steam, que vão ser emulados. E eu tenho um conjunto que vão ser enviados para um proxy, então: DNS e Memcached, que não seria viável gerar uma resposta toda vez que o honeypot receber requisição.

O sistema, ele é bem simples, ele recebe um proxy, ele tem um proxy que vai receber a requisição específica, ele vai criar um dicionário

em memória com um contador, para a vítima em específico. Como eu sou um refletor, o IP que eu estou recebendo está 'spoofado', então eu não tenho o endereço do bot, eu tenho o endereço da vítima. Eu vou verificar se eu devo responder ou não, então nós temos um *rate limit* para não gerar tanto volume de tráfego, como é o esperado pelo atacante. Caso eu não tenha batido nesse *rate limit*, eu vou resolver essa requisição e responder para a vítima. No caso, terminando o ciclo de um ataque específico.

O honeypot em si, ele vai salvar o ROA data desse tráfego de rede, então eu vou ter todo o *backup* desse tempo, como ele também vai ter um contador em memória, identificando os flows que estão sendo utilizados para ataque, como também payloads específicos. Que é o que eu vou acabar apresentando.

Então, o mecanismo de contenção. Em decorrência do grande volume de tráfego que um refletor vai gerar, nós fomos obrigados a definir, desde o início, um limite diário de requisições. Então, considerando trabalhos anteriores nossos, a gente definiu um limite fixo de cinco requisições. Com o passar do tempo, até nem é necessário cinco requisições, um conjunto menor seria adequado.

Também considerando o conjunto de scans que tem na Internet, nós tivemos que considerar bloquear a maioria. Então, nós definimos uma lista de blacklist, que basicamente consiste de um compilado de diversas fontes da Internet de scans, de várias organizações que estão scaneando diariamente a Internet, procurando refletores.

A implementação é bem simples, então, é um sistema que roda em Python, usa um banco de dados SQLite, ele não gera um conjunto de dados tão elevado para necessitar de outro tipo de banco de dados. Como alguns protocolos fazem proxy, eu tenho unbound e o Memcached instalado, para receber essas requisições de cache. O sistema, ele não precisa usar quase nada de processamento, tanto que ele tem um AMD Phenom II, 4 gigas de RAM. A máquina está localizada, então, na rede da UDESC, ela tem um IP global roteável e ela está coletando dado desde setembro de 2018. Então, ao todo nós tempos pouco mais de 731 dias de tráfego de DRDoS coletados pelo honeypot.

É bom deixar claro que quando o sistema foi colocado online, demorou algumas horas para ele começar a ser utilizado para ataque. Nossas experiências anteriores demoravam alguns dias, no MP em específico não precisou tanto tempo, em algumas horas o sistema já estava sendo explorado.

Só que nós temos um problema a partir do ponto que a gente vai fazer a avaliação desses conjunto de dados, nós temos um refletor. Nós não somos uma vítima, nem um atacante, nós estamos no meio termo. Então, nós precisamos avaliar o que seria um ataque e o que

seria uma vítima em específico. E, para isso, nós temos que identificar o que seria esses elementos, sendo que na literatura não existe uma definição específica, o que seria uma vítima para um carpet bombing.

Então, nós definimos que um ataque DRDoS, ele é formado por um conjunto de, no mínimo, cinco requisições com o mesmo IP de origem, para a mesma vítima, no caso, em um intervalo máximo de 60 segundos. Como eu vou ter vários fluxos para diferentes protocolos, essa definição foi adequada. E para uma vítima em específico, a gente acabou definindo que é os três primeiros octetos de um endereço IP de origem, então, bloco CIDR /24.

Agora, as nossas observações. Então, o que... que tipo de dado a gente acabou coletando. Então, ao longo desses 700 dias, a gente teve um volume total de tráfego de 1.8 terabytes. O que gera 20 bilhões de requisições, que é um volume relativamente alto, gerando basicamente, por dia, uma média de 28 milhões de requisições que eram utilizadas em ataques.

É legal deixar claro que como nós temos um *rate limit*, somente 0,034% foram respondidos, o que ainda é alguns milhões, 7 milhões, se não me engano. Ao todo, então, nós identificamos mais de 1,4 milhões de ataques e 1 milhão basicamente, 1,1 milhão de vítimas. Distribuindo esses ataques em monoprotocolo, então são os ataques que só exploram um único protocolo, então só vou explorar DNS, só explorar NTP, eu tenho a grande maioria. Então, 97% dos nossos ataques utilizaram só um protocolo. Mas nós tivemos um conjunto que utilizou mais de um protocolo, então, mais de um protocolo era utilizado ao mesmo tempo, para realizar o ataque.

É legal também ver a distribuição das vítimas. Essas vítimas, então, seguem a mesma distribuição, então o maior conjunto de vítimas e de ataques está concentrado no conjunto de monoprotocolos, mas ainda a gente tem um valorzinho legal de multiprotocolos, o que dá para fazer algumas avaliações com esse conjunto de ataque. Afinal, 13 mil, quase 14 mil ataques, não é pouca coisa. Essas vítimas em específico, elas estão distribuídas basicamente em Estados Unidos, China e Reino Unido. Nós temos algumas vítimas no Brasil, que eu vou entrar em mais detalhe depois.

Então, as intensidades desses ataques. A maioria dos ataques tem duração de pouco tempo, se for observar... tem duração de pouco tempo e tem um número de requisições baixo. Então, se eu for observar ataques monoprotocolos, que realizaram mais de um milhão de requisições, eu só tive 1.900 ataques. Que tinham mais de 1 milhão de requisições. Se eu for observar esse multiprotocolo, eu só tinha 60 ataques que tinham mais de 1 milhão de requisições. Os maiores ataques que agora são um pouco alarmantes. Então, o maior ataque

que nós tivemos, utilizando somente um protocolo, ele durou 15 horas; e para multiprotocolos, 11 horas.

Se a gente observar toda a distribuição, e esquecendo a parte de mono e multiprotocolo, dá para afirmar que a maioria dos ataques tem duração pequena. Então, a maioria dos ataques, 87% dos ataques não atingia 15 minutos; e 95, quase 96% desses ataques, não passava de uma hora.

Considerando agora, essa distribuição de monoprotocolo específico, que eu acabei de falar. Nós temos essa tabelinha linda aqui. Então, se for observar a distribuição, eu tenho uma distribuição bem baixa de protocolos que acabaram... de protocolos e ataques que acabaram sendo executados por mais de uma hora. Então, somente, nem 3% para monoprotocolo, mas eu tenho 12% para multiprotocolo. Isso mostra uma tendência, que multiprotocolos tendem a ter uma duração maior, superior a ataques monoprotocolos. Se eu pegar os ataques mais longos, eu não tenho essa... eu tenho ataques que duraram sete dias e basicamente para os dois ataques. Esses são os ataques mais... com maior duração.

Agora, como essa coleta foi realizada durante anos, é legal destacar a evolução temporal desses ataques. Então, esse gráfico mostra, esse *bar plot* mostra a evolução ao decorrer dos meses, dos ataques.

Então, no início nós tivemos um crescimento rápido. Mas diariamente, nós temos uma média de 2 mil ataques registrados no sistema. Eu tenho alguns meses que tiveram variações. Essa variação aqui, julho para agosto, essa variação em específico, distingue um conjunto de ataques que usou um número de requisições muito baixo. Então, tinha muitas vítimas que não eram do mesmo CIDR, que estavam recebendo poucas requisições. Como nós só temos um refletor, fica muito difícil tirar uma conclusão do que estava acontecendo nesse período.

Mais para o final, nos últimos meses de coleta, esse crescimento se deu pela adição de novos protocolos. Então o sistema é modular, então eu posso ficar adicionando protocolos específicos nele. Nesse período foi a adição do CLDAP e do CoAP, e como esses protocolos acabaram aparecendo na mídia nesse período, nós também conseguimos observar a crescente utilização desses protocolos.

Mesmo ponto agora, olhando as requisições que o sistema recebeu, grande parte do tempo nós temos a predominância de Memcached. Então, o Memcached apareceu em 2017, se não me engano... ou 2018. Eu não me recordo específico. Mas ele tem um alto fator de amplificação e ele vem, predomina em um conjunto de anos aqui. Então, dois, basicamente quase dois anos que ele predomina. Já em 2020, com a adição do CoAP e do CLDAP, nós percebemos que o

CLDAP, ele tem uma preferência pelos atacantes em relação ao Memcached. Claro, é possível ver que existem pontos que utilizaram DNS, Chargen e também CoAP, aqui no final. QOTD, desculpa. Não CoAP.

Olhando períodos específicos, então, nós temos a eleição presidencial de 2018. Nesse período, no mês em específico, a gente identificou que dobrou o número de pacotes que o sistema estava recebendo naquele mês. E no dia do segundo turno, em específico, nós tivemos um aumento de 227% no número de ataques.

Observando o período de *lockdown*, então, agora já em 2020, nós também conseguimos detectar algumas semelhanças às estatísticas que eu mostrei lá no início. Então, comparando o *lockdown*, o mesmo período, março a junho de 2019 e quatro meses antes, é possível identificar que teve um crescimento de 4% na taxa de pacote que o sistema estava recebendo. E também foi identificado o surgimento de novas vítimas, principalmente em organizações de saúde e comércio eletrônico.

Agora, o que é interessante identificar, é a utilização desses protocolos. Primeiramente, eu vou discutir os ataques olhando só em um protocolo, depois eu vou discutir o conjunto desses protocolos. Então, se eu observar especificamente essa tabela aqui, eu estou distribuindo os protocolos por ataque e requisições. Então, eu tenho predominância do Chargen e Memcached, tanto para ataques quanto para requisições. Mas é legal destacar que o CLDAP só está há três meses no conjunto de coletas e ele já está no nosso top 3, então ele tem uma vasta utilização. O Chargen e o Memcached já tinham predominância ao decorrer dos anos, mas os últimos meses, o CLDAP está sendo muito mais utilizado que ele.

Observando o fator de amplificação, um negócio legal de se comparar com outros trabalhos na literatura, a nossa amplificação... o fator de amplificação é inferior à maioria dos trabalhos que a gente encontra. Então, eu estou apresentando a média, alguns... do nosso sistema. A maioria desses trabalhos ou apresenta a média ou o valor máximo. Por exemplo, o Memcached aqui é o valor máximo que ele pode ser explorado. O nosso único caso que foi superior, a taxa de amplificação, foi a SSDP, que a nossa resposta em si era maior do que os outros trabalhos. Nós tínhamos... estava gerando dois campos que não eram necessário, e já foi retirado.

É legal destacar, então, que o Memcached tem, mesmo com o valor médio não sendo tão alto como destacado na literatura, ele ainda é superior a todos os outros protocolos.

Agora, olhando os ataques multiprotocolos, então quando eu estou utilizando mais de um protocolo ao mesmo tempo, eu consigo ver a presença do Chargen. Então, eu tenho todos outros protocolos

que foi possível ver na evolução temporal, DNS, CLDAP, SSDP nem tanto, e Memcached, mas eu consigo ver que esses protocolos em si, estão explorando o Chargen como combinação. E o Chargen é um protocolo Legacy, ele não deveria ser encontrado atualmente na Internet, ele nem deveria estar aberto na Internet em si, considerando a funcionalidade dele. E essas combinações correspondem a 82% dos ataques, e com o conjunto das solicitações relativamente alto também.

Agora, olhando os payloads, então, que tipo de informação está sendo enviada. Dois conjuntos podem ser destacados, eu tenho um conjunto de protocolos que precisa interagir e mandar um payload válido para eu gerar amplificação, e tenho um conjunto de protocolos que qualquer tipo de conteúdo que ele envie, eu vou gerar uma amplificação, que é problema do Chargen e do CoAP. Qualquer tipo de conteúdo que ele recebe, ele vai gerar uma resposta.

A partir desse momento, os atacantes vão tentar maximizar o fator de amplificação. Então, eles vão mandar o menor payload possível. Então, 100% dessas requisições que a gente recebe no Chargen, ela só tem um byte. No QOTD é a mesma coisa, a maioria das requisições, acho que 99% das requisições só tem um byte ou dois.

Agora, existe um conjunto de protocolos que depende, então, da mensagem que eu estou enviando, que são os protocolos mais conhecidos. Então, SSDP, eu sou obrigado a enviar uma requisição M-Search. NTP, eu sou obrigado a mandar um Monlist, para retornar a lista de endereços que eu tenho em memória. CoAP e CLDAP, a mesma coisa.

Eu tenho já, protocolos que dependem um pouco mais, eu vou ter um pouco de variações no payloads, como o DNS. Então, o DNS, ele vai depender do RR, do Resource Record. Eu tenho, ao todo, 115 mil RRs distintos que foram observados no sistema, mas, se for observar esse conjunto aqui, esse top 6, ele corresponde a 34% de todos os RRs que a gente acabou [interrupção no áudio]. E se for [interrupção no áudio].

Outro ponto que é legal deixar claro, é essas... Voltei? Essas Queries, esses QTypes que são encontrados no topo aqui, elas não foram só identificadas pelo nosso honeypot. Três delas em específico, essas duas primeiras e essa quinta, já foram relatadas alguns anos atrás na literatura, e elas ainda são utilizadas.

E, por fim, o payload do Memcached, que é o mais intrigante. Ele tem duas opções que podem ser exploradas pelo atacante. Então, a primeira opção é a mais fácil, o atacante não precisa pensar em quase nada, ele só precisa fazer uma requisição com payload de status, então, o que vai me retornar condições do sistema, então, retornar um conjunto, umas lista de status do sistema, quantos bytes estão locado à memória, quantas threats estão rodando.

Essa lista em específico, foi usada bem no início, quando a gente lançou o sistema, mas ela tem um fator de amplificação relativamente baixo, 32. Já o outro ponto que eles acabam utilizando o sistema, é 'setar' uma informação em memória. Então, eu vou adicionar um payload de memória e eu vou adicionar uma chave, e o atacante vai ficar fazendo milhões de requisições só para essa chave. Então, a gente só precisa adicionar uma vez, esse conteúdo, e ele vai ficar só depois, consultando esse conteúdo. O que é muito bom, e 99% das requisições que a gente recebe, foram desse tipo.

Claro que olhar esse conteúdo específico não provou nada, porque a maioria desses dados foi aleatório. Então, não tem muito o que dizer.

Olhando a distribuição, então, entre países, como eu comentei antes, Estados Unidos, China e Reino Unido estão no topo. O Brasil, ao todo, considerando monoprotocolo e multiprotocolo, ele atingiu 3,4% de todos as requisições, então, um valor relativamente baixo. Ao todo, então, monoprotocolo foi quase o dobro, 226 países que foram atacados, em relação a multiprotocolo.

O tipo de ASN que estão sendo atacados, então, a maioria é provedores, provedores de nuvem ou provedores de serviços de Internet.

E observando o carpet bombing em específico, nós temos alguns comportamentos. Se observar a variação do nosso conjunto de dados, nem 6% deles fez parte do carpet bombing. Então, 5,7% eram carpet bombing. E o conjunto de ataques não chegou nem a 4%. Mas o número de vítimas não segue essa mesma tendência, 21% das nossas vítimas estavam presentes em carpet bombing. Esse ataque é mais difícil de ser detectado, porque eu vou ter uma distribuição, como comentei antes. Então, eu não vou ter o mesmo início de tempo entre as requisições no sistema, eu vou ter um conjunto que vai começar em tempos diferentes, elas vão se intercalar. Sobrepor, desculpa. E elas vão terminar, depois de um tempo específico. Nem sempre vão terminar no mesmo período.

Se eu... Pegando um exemplo bem básico, bem simples que nós recebemos. Então, nós tivemos um ataque que durou 14 minutos. Dois protocolos foram utilizados, Chargen e Memcached - como eu mostrei na tabela antes, é um protocolo que é favorável para esse tipo de ataque – 344 mil solicitações foram realizadas e esse conjunto de tráfego foi distribuído entre 43 diferentes hosts no mesmo CIDR. Então, eu tenho quase 8 mil pacotes indo para cada vítima, então eu estou distribuindo o meu tráfego entre a vítima. Se eu for ver uma média dessa distribuição, eu fico próximo de quase 10 mil requisições por host, olhando só os ataques carpet bombing.

Mas é possível apontar que uma pequena taxa do bloco é utilizada, então, na nossa observação, somente 6% do CIDR estava, realmente, sendo utilizado. Observando os ataques que utilizaram uma quantidade maior do bloco, então 50% mais o bloco, eles tiveram uma média de... uma média alta de ataque... de requisições ao todo, mas um conjunto muito baixo de requisições para cada host. O que acaba mostrando uma tendência desses ataques.

E, por fim, um ataque que a gente acabou identificando, é uma tendência. Então, essas vítimas que eram encontradas no carpet bombing, eles acabavam aparecendo em alguns casos, com antecedentes. Então, essas vítimas eram atacadas em períodos anteriores, antes de um ataque carpet bombing acontecer. Isso não acontecia em todos os casos. Foram alguns casos específicos que a gente encontrou essa tendência, e nós consideramos esse ataque como antecedentes.

Então, esse estudo, em específico, permitiu encontrar e identificar essas tendências em ataques DRDoS, especificamente na utilização de refletores. Muitos desses protocolos, eles são Legacy. Então, Chargen, QOTD(F), são protocolos que não deveriam estar abertos na internet. E tem protocolos em si, como o Memcached, que não deveriam também estar abertos, considerando muita coisa, mas ainda estão. E eles estão sendo utilizados com muita frequência, como é possível ver. O nosso ponto, nós só temos um ponto de observação, então, essas estatísticas tendem a ser muito pior se a gente tem um sistema um pouco maior. Então, vou ter um volume muito maior de dados, vou ter um conjunto muito maior de requisições e, em muitos casos, a distribuição do carpet bombing específico, pode ser maior entre as vítimas. Para o futuro, a gente está planejando aumentar o número de honeypots, justamente para permitir uma melhor avaliação desse tipo de ataque e comportamento.

Aqui eu tenho umas referências, se alguém tiver curiosidade, é mais na parte de amplificação. Seria isso, muito obrigado. Perguntas?

SR. RUBENS KUHL: Obrigado, Tiago. A gente tem algumas perguntas. A primeira foi do Fábio Aquino(F). Ele pergunta qual o papel real do refletor, o que ele faz, de fato, e o que diferencia do DDoS convencional.

SR. TIAGO HEINRICH: O DDoS convencional, eu não tenho um intermediário. Então, eu tenho um indivíduo que tem um conjunto de bots ou um conjunto de máquinas maliciosas, e esse conjunto de máquinas vai gerar tráfego e enviar para uma vítima. Quando eu coloco esse R a mais, então no DRDoS, eu tenho um cara a mais aí no meio, eu tenho um refletor. Então, esse refletor, ele vai pegar uma mensagem que o bot está enviando, então em vez de enviar essa requisição direto para a vítima, eu vou pegar essa mensagem... o

refletor vai pegar essa mensagem e gerar uma mensagem ainda maior. Então, como eu tenho esse conjunto de protocolos, a ideia deles é enviar uma requisição pequena para o refletor, gerar uma mensagem ainda maior, uma mensagem muito grande para a vítima. Como acontece em alguns casos, eu vou ter uma mensagem razoável do bot para o refletor, que pode ser enviada direto para a vítima. Mas vale mais a pena enviar essa mensagem para o refletor, que vai gerar uma mensagem ainda maior, e enviar essa mensagem para a vítima.

Qual é a vantagem de fazer isso? Eu não preciso de tanto bot. Então, no caso que eu usar dez bots, quando eu estou explorando um refletor, eu não vou precisar de dez. Às vezes com três eu vou conseguir valores ainda maior. Então, essa é a real vantagem da utilização de refletores. E como tem muito protocolo Legacy e muitos sistemas mal configurados em si, que podem ser usados como refletor. Eu listei nove aqui, tem uma lista na Akamai, eles fazem uma tabela periódica, de tanto que tem. Então, é muita vantagem de eu utilizar um refletor para os ataques. E a tendência é que esses ataques migrem para só usar refletores. Ao longo dos anos, eles tendem a utilizar, cada ano mais, só ataques refletores. E a maioria dos ataques, eles falam DDoS, mas se pegar a fundo e ver, por exemplo, do GitHub em 2007, se eu não me engano, aquele ataque era DRDoS, ele não era um protocolo só. A maioria dos ataques, ele vai utilizar mais de um protocolo. Eles falaram que era Memcached, mas tinha não sei quantos por cento de NTP ali no meio. Eu acho--

SR. RUBENS KUHL: A gente tem duas perguntas do Erinaldo Santos. A primeira é se, no caso aí de ataques relâmpagos, fica difícil de identificar o hacker. E o outro é se existe algum antivírus ou programa que possa rastrear, identificar e prevenir de onde partem os ataques.

SR. TIAGO HEINRICH: Então, nós não fizemos fingerprint de nenhum dos bots, nós não estamos mexendo com nenhum dos lados, então, o meu refletor está no meio. Eu não estou... quando eu recebo uma requisição, tanto de *scan*, que eles vão me descobrir na internet, quanto de uma requisição para reflexão, eu não estou fazendo *scan* para identificar esse elemento e nem a vítima. Então, nem o bot, e nem a vítima. Então, não. No nosso ponto de vista, a gente não tem nenhuma noção de quem seria o atacante. Nós temos uma noção de quem seria a vítima, que é o volume de tráfego que a gente está levando.

Já para a mitigação, a principal vantagem para mitigar esses ataques é se livrar dos protocolos. Então, não permitir que esses protocolos estejam abertos na internet. Como eu comentei do carpet bombing, é extremamente difícil a identificação de alguns desses ataques, porque tu não está atacando uma única vítima. Uma coisa é

tu identificar um volume alto de tráfego chegando para uma máquina, que a maioria das abordagens vai tratar disso. Então, eu tenho um ponto na rede, que eu vou observar se está passando um conjunto muito alto de tráfego. Eu vou começar a avaliar: ah, que tipo de conteúdo está passando ali? Então, nesse ponto da rede, eu vou tentar tratar esse ataque. Quando eu estou falando de um carpet bombing é mais difícil, porque eu estou distribuindo. Então, se torna muito mais difícil a mitigação.

E no nosso ponto de vista, a gente não está mitigando nada, a gente só está observando, a gente está fazendo o papel de um sistema que está sendo explorado. Então, a gente não está tentando... claro, a gente não quer prejudicar a vítima, mas a gente não está tentando parar o atacante também, em nenhum ponto.

SR. RUBENS KUHL: Obrigado, Tiago. De perguntas que eu tinha, é isso. Obrigado pela apresentação.

SR. TIAGO HEINRICH: Eu que agradeço.

SR. RUBENS KUHL: Adriano?

SR. ADRIANO CANSIAN: Legal, muito obrigado. Muito bacana essa discussão, e a gente até gostaria de falar mais sobre isso, não é? Mas estamos em cima da hora aí. Então, muito obrigado novamente. E nós vamos passar diretamente, então, a próxima apresentação.

Mas antes disso, eu gostaria de lembrar que os materiais, os slides estão disponíveis via protocolo FTP, no site: ftp.registro.br/pub/gts/gts35. Você vai encontrar fácil. Se você tiver dificuldade com FTP, né, se você é um cara mais moderno, tal, gosta de usar https, é só você ir no seu navegador e trocar o FTP, o protocolo, claro, né, e acessar o mesmo site: ftp.registro.br/pub/gts/gts35. Se você não sabe usar FTP, é uma boa oportunidade para aprender.

Bom, muito bem, então, em seguida vamos dar as boas-vindas ao Dioraci Corrêa Junior. Dioraci Corrêa Junior é analista no Banco do Brasil e vai falar para a gente sobre o PIX e seus desafios de segurança. Dioraci, seja bem-vindo.

SR. DIORACI CORRÊA JUNIOR: Bom dia, pessoal. Tudo bom? Bom, eu preparei uma palestra aqui, mais voltada para a prática, menos em detalhes em níveis de protocolo, que é o que acontece no nosso mundo real, no nosso dia a dia, tá? Deixa eu passar o slide aqui.

Uma agenda. Um pouquinho de história, do que é o PIX. Os desafios que esse produto veio a adicionar para a gente, na indústria bancária, tanto na indústria de TI, bancária, quanto segurança bancária. Os próprios desafios de segurança e também uma conclusão.

Bom, vamos lá. Um pouco de história, não é? O que é o PIX? É um meio de pagamento instantâneo. Ele é uma demanda criada pelo

Banco Central brasileiro, Banco Central do Brasil, Bacen. Através dessa Circular 4027, que publicada em junho passado, agora, dia 12 de junho, que institui esse produto bancário, vamos colocar assim, de meio de pagamento, transferência de recursos e pagamento de recursos entre bancos, interinstituições. Pode ser intra e interinstituições, não é?

E de onde veio isso? Como alguém criou isso? Na verdade, esse negócio tem uma história até, imagino, longa no mundo aqui. Que existem outros países que já têm mecanismos de pagamento, ou meios de pagamentos instantâneos. Alguns *cases* bem grandes e bem-sucedidos, vamos colocar assim, que estão espalhados pelo mundo. Tem esse da Índia, que é o UPI, *Unified Payments Interface*, que existe desde o mês de abril de 2016. Na China... Esse UPI, ele tem cerca de 300 milhões de transações/mês, em volume, hoje, na Índia, que é um país com capital bem baixo, assim, em termos de ativo circulante. O WeChat, que aí, dentro do próprio WeChat, que é o concorrente chinês do WhatsApp, eles criaram o WeChat Pay, que é um meio de pagamento, que você paga por QR Code, você paga, transfere dinheiro entre instituições, instantâneo. Também instituído em 2016. O WeChat tem aproximadamente 1.1 bilhões de usuários, com 900 milhões de usuários mensais ativos do WeChat Pay. E tem uma iniciativa também no México, mais recente, chamada CoDI, que é o... instituído em setembro de 2019, também no meio de pagamento.

E para que ele serve? O que é o PIX? Como o próprio nome diz, é para fazer um pagamento instantâneo. Enviar e receber dinheiro de ou para uma conta sua, ou de um terceiro, ou de uma empresa. O sistema brasileiro financeiro, ele estava carente de ter um produto desse tipo, porque, a meu ver, assim, analisando esse mercado bancário, nosso sistema de pagamento, nosso sistema brasileiro bancário é um dos mais evoluídos do mundo. A gente tem transações on-line que a maioria dos sistemas bancários não têm. A gente faz transação on-line como se fosse uma coisa comum para a gente. Mas, fora desse mercado, olhando para fora, por exemplo, Estados Unidos. O sistema bancário dos Estados Unidos hoje, raramente você consegue ter uma transferência on-line. Ele é feito basicamente em cima de cheque. Então, o cara vai pagar uma conta, ele pega um cheque, bota no correio, três, quatro dias a conta está paga. Ele vai transferir um dinheiro de um banco para o outro, ele manda uma ordem para transferência, aí o cara emite um cheque, e vai para o outro lado, aí o banco compensa. Então, assim, é uma coisa que ainda está em um modelo bem antigo, apesar de eles serem um país de primeiríssimo mundo, não é? Então, o sistema bancário deles ainda é baixo. Mas eles já têm iniciativas em meios de pagamento, principalmente com empresas privadas, tem N, Apple Pay, Google Pay, N outros pays que

você consegue fazer isso aí, mas nenhum vinculado a um sistema bancário mesmo, é um produto de terceiros.

E o nosso sistema financeiro, ele vem evoluindo durante esses anos, criando novos pagamentos, novos produtos que venham a modernizar esse cara ainda mais. E o PIX é um deles. Uma das coisas que eu cito aqui, que é muito interessante, é que recentemente foi... recentemente assim, né, tem uns anos aqui, vamos colocar assim, que foi regulamentado pelo regulador, as instituições de pagamentos, as IPs, e não só as IFs, que são as instituições financeiras, que são os bancos comuns que a gente conhece, não é? Então, isso fez com que o mercado fosse acelerado nessas questões de tecnologia de pagamento e tudo mais, onde vieram grandes *players*, como Nubank, PicPay, Mercado Pago, e tem N outros *players* aí, que fazem essas instituições de pagamento também. E isso é uma evolução no sistema financeiro, eu vejo isso com bons olhos.

Tá, e quais são as vantagens do PIX? Por que esse cara tem que ser feito. São transações feitas de forma on-line, instantâneas e 24/7. Tem redução no custo final para o cliente, porque isso, pela regulamentação, pessoa física não paga para emitir um PIX ou para usar a ferramenta PIX. E ele veio, como eu falei, preencher uma lacuna existente nessa cesta de instrumentos do mercado bancário que a gente tem hoje. Incentivando até o uso de dispositivos eletrônicos desse mercado, não é?

Legal, mas veja bem, é muito bom, é cheio de vantagem, mas ele é um produto completamente novo, com uma implementação bastante disruptiva, do ponto de vista de infraestrutura, de regulação e consumo, por quê? Já existia. Você fala: Ah, Dioraci, mas existe TED, existe DOC, você consegue transferir dinheiro entre um banco e outro de uma forma normal e até rápida. Sim, existe. Porém, esse cara aqui mudou muito a forma como isso era implementado. Porque esses ativos DOC, TED, essas transferências DOC, TED, pagamentos, eles usam uma sistemática que a gente chama de arquivos. Você monta um arquivo, junta-se um pedaço do tanto de TEDs que precisam ser enviados para uma instituição, fecha-se um arquivo e ele vai ser processado na outra instituição.

O PIX acabou com esse tipo de coisa. Então, a implementação dele é uma implementação que ele é on-line mesmo. É transação uma a uma. Eu enviei um PIX lá para o Adriano. Ele vai lá no banco do Adriano e credita o banco. Não tem arquivo, não tem nada. É uma transação em *real time* mesmo. Um dos grandes desafios para esse cara é que ele é um produto multicanal. Então, o PIX, ele pode ser implementado, pode ser emitido ou usado em muitos canais de autoatendimento dos bancos. Então, eu posso ter PIX dentro do meu internet banking, eu posso ter PIX dentro do meu mobile, do meu

aplicativo móvel, eu posso ter PIX dentro de um WhatsApp, um chat. Posso ter PIX em um caixa eletrônico, eu posso ter PIX dentro de uma agência bancária, como é o caso do Banco do Brasil. Então, assim, o desenvolvimento disso durante esse curto espaço de tempo, como vocês viram, foi publicado, a versão que instituiu ele, em 16/6, para entrada em produção em 3 de novembro. Então, de junho a novembro, corre, tem que rodar, e é demanda regulatória. Então, isso é um belo de um desafio com relação a isso, como fazer isso.

E aí, um dos desafios grandes em cima desse cara, é um desafio para a segurança. Por quê? Conscientização dos usuários. É difícil você colocar um produto novo na rua e garantir que os usuários, ou quase impossível garantir que seus usuários vão entender o que é esse cara e vão entender que eles podem ser vítimas de fraude, vão entender que eles podem ter o nome desse produto usado de uma forma para tentar burlar alguma coisa. Então, esse, para mim, ele é um dos desafios se não o maior desafio com relação ao PIX, ou ao mercado bancário, esse novo produto. Que é um produto novo para o mercado e é novo para o consumidor também.

E isso gera o que eu falo, a farra da engenharia social, não é? Toda vez que a gente tem um lançamento de algo grande, ou da magnitude que é, por exemplo, o PIX, cara, isso vira, massificadamente, uma enxurrada de tentativa de golpes.

Então, eu trouxe uma notícia aqui, de um site: PIX vira um novo desafio para combater golpes na internet. Então, assim, aí eles destacam nessa matéria que, assim, a falta da maturidade digital do brasileiro. A capacidade de organização do cibercrime, isso instituiu uma forma ampla e irrestrita de tentativa de golpe. Então, aqui nessa matéria, até tem o link aqui no material, eles falam que com um dia de lançamento... porque o PIX, ele foi implementado em fases. Então, eu tenho uma primeira fase para abertura para o cadastramento das chaves PIX, para o cara se habilitar a receber um PIX, né, o cliente se habilitar a receber um PIX. E, em seguida, meses depois, um mês depois, vinha efetivamente, a abertura para as transações, para início das transações. Um dia após o registro da abertura, né, das chaves, do cadastramento das chaves PIX, já tinham mais de 70 domínios falsos registrados com tentativa de golpe.

Aí entra... começa, chavepix.me, gerenciadorpix. Como o usuário, ele não tem esse conhecimento, ele não consegue absorver esse conhecimento de uma forma rápida, isso aqui virou um golpe bastante frutífero, não é? E isso vem enviado por SMS, WhatsApp, e-mail, Telegram, qualquer coisa que o cliente, ou o usuário tenha acesso, isso, ele vai receber algo nessa linha. Ele pode estar sendo alvo de uma tentativa de engenharia social. E como o Adriano sempre dizia, isso é muito barato, você fazer um ataque desse tipo.

Aí, eu tenho até um 'casezinho' aqui do "*click here for doughnuts*!", não é? De algo que aconteceu comigo. Isso aqui, ele não está associado diretamente a um fluxo do PIX. Isso é um *print* da minha tela de celular. Eu estava fazendo um teste de uma liberação de um dispositivo, recebendo um SMS de retorno, tal. E minutos depois, eu recebo um SMS de um número totalmente diferente, com um link para clicar, referente ao cancelamento do meu... expirou o cancelamento do seu mobile. Assim, para um cliente leigo, é plenamente inteligível. O cara acabou de fazer uma liberação de um dispositivo através de um coisa, e ele recebe em seguida, um pedido do... que iria cancelar aquilo. O cliente vai clicar. E aí, quando clica, começa. Barra da página falsa, de pedir informação. Aí vai aqui, agência, conta, senha. Número do celular, a outra senha do cara. Número do cartão de crédito, o código de segurança do cartão de crédito, o cara pede... literalmente, eu brinco que ele pede até a cor da cueca do cara. E aí, vai. O cliente vai entregar.

E isso, a operação realizada com sucesso. Isso acaba gerando para esse tipo de atacante, insumos para que ele intensifique outras fraudes. Então, hoje, com esse tanto de links maliciosos, de páginas falsas que os clientes realmente caem, ele, o fraudador consegue obter informações lícitas do cliente, informações quentes e reais que isso fatalmente volta para o próprio cliente, porque entra com... potencializa outros tipos de ataques. Ataque de falsidade ideológica, o cara pode usar esses dados que o cliente passou para abrir conta em outra instituição, ou para tentar fazer um golpe usando o nome da pessoa que está lá. Porque ele tem os dados reais do cliente, o cliente informou. O cliente, incrédulo, ele acabou informando, não é?

E esses ataques acabam ficando mais direcionados, porque esses ataques de engenharia social, com esses links, com essas páginas e tudo mais, são massificados, são ataques, literalmente, no atacado, e não no varejo. Só que uma vez que o cliente cai em alguma coisa desse tipo, eu posso entrar para um ataque no varejo, que é um ataque mais elaborado. Porque o fraudador vai ter acesso aos dados desse cliente, as informações pessoais desse cliente, e pode lançar um ataque direcionado em cima do cara. É muito comum ligarem para ele, é muito comum mandarem e-mail ou botar alguma coisa bem direcionada, com dados reais do cliente, que dá legitimidade ao ataque. E forçar o cliente a se detonar até mais, entendeu? Porque o cara vai, ele vai entrar, ele vai entregar, realmente, ele vai ter uma informação lícita, do cliente. E isso acaba, cara, virando uma bola de neve, não é?

Golpes mais comuns. Falsos... voltando ao PIX, não é? Falsos cadastros dessas chaves PIXs, sequestro de contas em aplicativos de mensagens. Assim, é muito comum a gente ver: 'Pô', sequestraram meu WhatsApp, estão pedindo dinheiro por contato. Sequestraram a conta do Facebook, estão mandando mensagem no Messenger, e tudo

mais. Então, assim, isso acontece diariamente. E, cara, por melhor que o cara seja instruído, dependendo da forma como está, do dia que ele está, esse cara, ele vai cair. Um golpe que assim, é extremamente comum, é o golpe de sites de venda. O cara faz um anúncio em um site de venda, bota o número do celular dele lá como um contato, "Estou vendendo meu PC usado aqui", e tal. O 'mala' vê aquilo lá, o fraudador vê aquilo, liga para o cara: "Ó, aqui é do site tal, você fez um anúncio? Eu preciso confirmar alguma coisa com você, tal. Eu vou te mandar um código aí." Aí prende a atenção do cara no telefone e manda um código. Só que o código não é do site de venda, o código é de uma transferência de WhatsApp, é um segundo fator de um banco, é uma liberação de um dispositivo, é alguma coisa que o cara vai usar ilicitamente.

E o cliente, no telefone, preso com atenção, ele acaba encaminhando aquilo para o fraudador e aí, sequestra conta, aí começa todo o problema na vida do cliente. E o cliente achando que ele estava falando com alguém lícito, de um site de vendas ou de algum anúncio, entendeu? E isso é uma coisa que é o dia a dia, não é? Com o incremento da pandemia, devido à pandemia, né, a gente vê claramente nos canais, uso de canais, assim, um incremento grande. E isso é preocupante. Por quê? Quando você coloca pessoas novas em uma situação forçada, a pessoa acaba não tendo um tempo de se interessar por aquilo, de se preparar para aquilo, de se resguardar com relação a isso. Porque o mundo virtual é a mesma coisa do mundo físico, o mundo físico, você está sujeito a ser assaltado, a ser atacado, alguma coisa assim. Só que a pessoa não imagina que isso vai acontecer. E aí, quando você coloca uma série de pessoas que nunca tiveram contato com esses tipos de ataques, com esses tipos de dispositivos, para dentro de um aplicativo móvel, de uma conta ou de uma modernidade dessa, essa pessoa é alvo fácil. Ela acaba se tornando um alvo fácil para esse tipo de investida. E isso, a gente vê, exponencialmente, isso crescendo nos últimos meses. Essa quantidade de pessoas que estão entrando para esse mundo virtual, que estão sendo meio que compelidos a utilizarem aplicativos de banco, aplicativos de mensagem etc., por conta da nossa situação. E para o fraudador, isso também é terreno vasto, não é?

Aí, eu inseri aqui, a pausa para um comentário, isso é até interessante. Houve uma... eu recebi essa notícia em 'trocentos' grupos de WhatsApp, de mensagem, de *timeline* e tudo mais, de um ex-diretor do Banco Central que declarou que o PIX é extremamente frágil em segurança, "Eu não vou usar", "Eu não quero o PIX". Porque, pela tese dele, com o PIX eu poderia ser sequestrado na rua e o cara poderia me forçar a fazer uma transferência, e limpar a minha conta ou fazer alguma coisa do tipo, um ataque físico. Só naquela prática, na prática, hoje isso já acontece se você pesquisar. Porque uma TED, um

pagamento, uma transferência entre contas, ela é feita de forma on-line. Então, o PIX não veio a causar um tipo de problema a mais. Ele é mais uma facilidade em cima de algo que já existia, e tende a melhorar ainda, por conta do meio de pagamento. Porque ele tem a parte de você conseguir pagar com QR Code, fazer transferência para empresas e tudo mais. E aí, a pessoa vem e causa um pânico desse tipo: "Não, porque eu não quero PIX, porque isso não é...". Como se diz? "Não é seguro, o cara pode me abordar".

E a gente vê que, por experiência até, na indústria de fraude bancária, que geralmente, o fraudador físico, o atacante físico, ele não é um atacante virtual. O atacante virtual, ele está não se sabe onde, o cara não aparece, ele está fazendo um ataque de forma massificada de longe. Mas o fraudador físico não. O cara que é... ele não conhece. E aí, ele não conhece o porquê ele está atacando, ele quer dinheiro. Ele vai te atacar para fazer sequestro relâmpago etc. Então, isso já acontecia. E essa tese de qualquer implementação que você faça em termos de segundo fator ou de novos produtos que vai direcionar para o mundo físico, ela sempre pode ocorrer, mas raramente isso acontece. Eu, particularmente, não tenho reporte, eu não conheço nenhum caso desse tipo de coisa que aconteceu.

E isso foi, assim, na indústria bancária, toda vez que a gente tem uma implementação de algo assim, acontece isso. Ah, quando foi colocado o cartão com chip, impediu clonagem de cartão. "Ah, então agora o cara vai me sequestrar para ter o chip do meu cartão". Não, não acontece. Hoje, todo mundo tem cartão com chip e isso não é um golpe que é frutífero, não dá tanto dinheiro assim. Então, fica no descaso. Quando a gente partiu para tecnologias biométricas, colocar biometria para saque, essas coisas. "Ah, o cara vai cortar meu dedo para levar lá no terminal, para sacar meu dinheiro". Não, não vai, porque não funciona, não é assim. A pessoa tem que estar viva, tem toda uma coisa assim. Então, quando alguém solta, alguém com grande influência fala uma frase dessa, causa mais terror do que na realidade, porque não é bem por esse lado que a coisa funciona. Mas isso foi um parêntese à parte aqui.

Então, outro grande desafio que a gente teve para segurança, está muito vinculado às características do PIX, que são transações feitas de forma on-line, eu estou conectado e vendo a coisa rodar. Com um SLA curto, então, o produto em si, PIX, ele tem um SLA de 10 segundos para conclusão de uma transação. E esse SLA é medido, se eu não fizer, cabe até penalização e multa. Vinte e quatro horas por dia em sete dias na semana. Isso que não existia no mercado bancário. TED, DOC, essas coisas, eram tudo de segunda a sexta, horário comercial, depois disso, acabou. Então, assim, isso mudou o paradigma de como o mercado bancário olha essas coisas. Então, monitorar essas transações... por exemplo, a gente tem, toda

instituição financeira tem o monitoramento transacional, que tem perfil de clientes e tudo mais. Então... Só que esse cara, a contramedida desses monitoramentos, ele ainda tinha um certo tempo. Porque, "Ah, eu vou fazer um DOC. Ele vai para compensação. Se der algum problema, eu consigo ainda rever, eu consigo parar", alguma coisa assim. Com o PIX não tem isso. É on-line. Foi, foi. Se o fraudador ganhou uma conta e mandou dinheiro, se você não conseguiu segurar, impedir a autorização desse cara. Acabou, o dinheiro foi embora, o prejuízo está feito.

Então, isso foi um desafio muito grande para a gente, para reconstruir, construir um monitoramento de forma que eu consiga ter esse perfil do cliente traçado, sem causar nenhum tipo de impacto, principalmente para o cliente. Porque a gente brinca, não é? O monitoramento, a camada de monitoramento, ela é o último muro antes de o cara entrar na sua casa. Se ele passou, já era. Só que se o muro está tão alto, ninguém consegue entrar. Aí eu começo a apertar e penalizar o cliente. É muito fácil eu não bater meu carro, é só não tirar ele da garagem. Então, assim, se eu começar a barrar e causar impacto no cliente, isso também não é legal, não é? Então, isso foi um grande desafio. Atender a essas características do produto no mercado bancário é uma coisa bem nova e bem inovadora para a gente.

Bom, aí eu fiz uma apresentação mais curtinha. Conclusões: os pagamentos instantâneos... E sim, eles eram uma lacuna no nosso sistema bancário brasileiro. A gente não tinha algo multi-instituições e de forma amplamente utilizada como existiam em outros países. E apesar de desafiador, é bastante revolucionário, ao meu ver, e bastante seguro também. Porque acaba criando alguns mecanismos, algumas coisas que a gente teve que evoluir para melhorar a segurança do mercado bancário como um todo. E forçou a aceleração nessas questões de segurança, de evoluir, tanto para o usuário, para o usuário entender que aquilo funciona, e também para que os próprios *players* se adaptem a esse tipo de coisa.

Porque eu, particularmente, não me lembro, eu tenho 15 anos de banco, já, eu não me lembro de nenhum produto que foi criado de forma tão grande assim, de tão... um movimento tão grande e tão rápido de ser feito com tanto impacto para o cliente final. E existem, sim, criações de alguns produtos que foram feitos, de cobranças e tudo mais, mas eles são produtos que possuem nichos bem determinados, empresas, pessoa física, crédito, alguma coisa assim. Mas um produto que é amplamente utilizado para qualquer usuário final? Foi a primeira vez que a gente viu. E isso é bem revolucionário. E considerando essas questões, eu acho que, assim, é um grande sucesso, a gente viu, acompanhando agora o último mês das implantações, da implantação do PIX, rapaz, o volume transacional está sendo bem interessante, bem... bastante pessoas usando, mas muita gente mesmo. E eu acho

que isso tende a expandir cada vez mais e a gente ter algo sendo massificadamente utilizada, não é?

Beleza. Aí eu deixo uma frasezinha que eu sempre brinco com os meus meninos lá: *"Todo dia sai de casa um bobo e um esperto. Se os dois se encontrarem, dá negócio."* Então, obrigado aí, gente. A gente pode abrir para as perguntas aí.

SR. ADRIANO CANSIAN: Obrigado, Dioraci. O Rubens vai fazer algumas perguntas em seguida. Por favor, Rubens.

SR. RUBENS KUHL: Obrigado, Dioraci, obrigado, Adriano. A primeira pergunta vem do Erinaldo Santos, ele cita que o Brasil é um dos países que está comtemplado, né, com o sistema de pagamentos instantâneos. Ele pergunta se a gente está atrasado em relação a esse produto que deveria ter mais segurança.

SR. DIORACI CORRÊA JUNIOR: Cara, atrasados eu não digo que estamos. Estamos, assim, como eu disse existem alguns países que já têm esse tipo de [ininteligível] fática de pagamento instantâneos. O Brasil ainda não tinha. Mas, assim, frente a outros mercados grandes, por exemplo, a maior economia do mundo, uma das maiores economias do mundo, Estados Unidos. Estados Unidos não tem isso. Canadá não tem isso. Grandes países que movimentam cifras 'trilhonárias', digamos, todos os meses não têm isso ainda. Então, assim, eu vejo que poderia ter sido antes? Poderia. Vimos Índia, China com coisas desde 2016. Mas o Brasil, eu acho que ainda está em um time bem legal, assim, a gente consegue expandir isso e usar e colher bastante benefício com relação a isso.

SR. RUBENS KUHL: Legal, próxima pergunta que temos do [ininteligível]. Ele pergunta se o PIX seria um tipo de criptomoeda.

SR. DIORACI CORRÊA JUNIOR: É mais ou menos, é parecido com o modelo de criptomoeda. Porém tem centralizador, orquestrador, vamos colocar aqui... muda o que chama de meio de liquidação. Criptomoeda tem blocos, falar em blockchain, mineradores, blocos centrados e não tem figura centralizada na liquidação dele, que vai fazer esse controle, esse... ela é distribuída, uma estrutura distribuída, não é? E o PIX não, ele é meio de pagamento, o meio de liquidação é o Banco Central. Então, eu tenho um regulador que coloca isso... O pessoal pediu para interromper as respostas.

SR. ADRIANO CANSIAN: Olá, pessoal. Desculpem aí pela interrupção, tivemos um problema técnico. E nós vamos retomar, então, onde nós paramos, com as perguntas, a partir da primeira pergunta. Rubens, por favor.

SR. RUBENS KUHL: Retomando, então, a nossa primeira pergunta é do Erinaldo Santos, ele comentou que o Brasil é um dos países que agora está comtemplado com o sistema de pagamentos

instantâneos. E ele pergunta se estamos atrasados em relação a esse produto que deveria ter mais segurança.

SR. DIORACI CORRÊA JUNIOR: Bom, vamos lá. Cara, eu não diria que nós estamos atrasados. Existem alguns países, grandes economias mundiais que ainda não têm um sistema da magnitude do PIX e de pagamentos instantâneos. Claro, existem outros *players*, por exemplo, a China tem, Índia tem, que são países gigantescos, têm populações gigantes que usam muito isso, mas o Brasil ainda não. Porém, com a instituição agora do PIX, eu imagino que assim, poderia ter sido mais rápido, poderia ter tido alguma coisa anteriormente? Poderia, mas, também, eu não vejo como fora do *time*, não vejo como um atraso com relação a isso. Muito pelo contrário, eu acho que é muito bem-vindo, e em um *time* até legal, talvez, até, anteriormente não teria tanto sucesso como está tendo atualmente.

SR. RUBENS KUHL: A próxima pergunta é do Tulio Adriano Muniz, ele pergunta se o PIX seria um tipo de criptomoeda.

SR. DIORACI CORRÊA JUNIOR: Ah, ok. Boa pergunta. Na verdade, o PIX se assemelha bastante com uma criptomoeda, porém, como a gente diz, ele muda o meio de liquidação. Uma criptomoeda, eu não tenho uma figura de um agente central fazendo a liquidação da intermediação financeira. Ela é feita através dos blocos, dos mineradores, da estrutura de blockchain convencional, por exemplo. Da qual existem entidades distribuídas que certificam aquilo, de que aquele recurso foi transferido. O PIX a gente tem um regulador e tem um centralizador com relação a isso, que é o Banco Central. Então, o Banco Central, o meio de liquidação é através da câmara dele, do sistema dele. Então, ele tem esse controle de onde veio e para onde vai toda a intermediação financeira do PIX. É semelhante a uma criptomoeda, porém não é o mesmo meio de uma criptomoeda. Ela tem um regulador que olha por isso.

SR. RUBENS KUHL: A próxima pergunta é do João Petronio Martins, ele pergunta se ele pode adicionar o PIX no sistema dele. Eu já adianto, Dioraci, que a gente já postou no chat o link para uma dessas formas que é a API PIX, a gente postou o GitHub do Banco Central, mas você pode complementar.

SR. DIORACI CORRÊA JUNIOR: Isso. É, o PIX, ele é um produto bem grande, bem amplo. Tem a API PIX, para fazer chamada direto, e tudo mais. E ele tem até uma modalidade que a gente chama de participante indireto. Que aí uma empresa, por exemplo, eu quero instituir PIX para os meus clientes. E eu não quero ir ao Banco Central, ter todas as travas e regulação. Então, existe a possibilidade de eu ser um participante indireto, da qual quem vai fazer essa ponte com o Banco Central é uma instituição financeira que hoje já opera e eu faço a minha conexão com essa instituição, que aí eu viro um participante

indireto. Então, por exemplo, eu crio a minha empresa, meu aplicativo aqui, eu quero instalar a chave PIX para o cara mandar dinheiro para a minha carteira, para alguma coisa assim, e eu posso ir, por exemplo, ao Banco do Brasil e falar: Banco do Brasil, eu quero participar do PIX, mas eu vou ser um participante indireto, você faz a interface com o Bacen para mim, faz toda essa parte contábil e toda a intermediação e eu falo com você. Só que aí eu... teoricamente, o participante indireto, é o nome dele que está lá, então, é um cliente dele, não é do banco. Ele só faz essa intermediação, ou uma outra empresa pode fazer essa intermediação.

SR. RUBENS KUHL: A gente teve também no chat várias interações sobre questões que não são sobre segurança de informação no PIX, sobre se o PIX é vantagem para o banco, o cliente, tarifa... São questões bastante controversas aqui, mas não relacionadas ao tema do evento. Mas que a gente já notou que várias pessoas já responderam, interagiram, então a gente entende elas como endereçadas.

E voltando à segurança da informação, o Erinaldo cita que ele já foi hackeado várias vezes na conta de e-mail, rede social, conta pessoal da Caixa Econômica e também no auxílio emergencial pelo app Caixa Tem sem o consentimento dele, como se faz? E se com a chegada do PIX não iremos ficar mais vulneráveis a ataques hacker.

SR. DIORACI CORRÊA JUNIOR: Cara, não ficaríamos. Eu entendo que não, tá? Porque o PIX, na verdade, ele é mais uma transação dentro de um ambiente 'logado', de um Internet Banking ou de um aplicativo de bancos, como se fosse uma outra transação comum. Ele adiciona a facilidade para o usuário final, porque ele é prático com relação a quebrar as amarras entre instituições e com a possibilidade, inclusive, de pagamentos via QR Code. Então, o cara, por exemplo, o tiozinho que vende pipoca lá na frente da missa, lá. Ele bota um QR Code lá, da pipoca a R$ 5,00 dele. Você vai lá, escaneia aquele QR Code e paga ele, on-line. Mas isso é uma transação em um ambiente 'logado'. Ele não tem um diferencial com relação ao que existia antes.

Então, eu entendo que com a entrada do PIX, eu não tenho um incremento de problemas com relação a esse tipo de... para o usuário final. Do tipo: Ah, o cara vai me hackear? Vai conseguir fazer alguma coisa mais fácil? Não. Eu não entendo dessa forma. Porque é mais um produto dentro do rol do leque bancário à disposição para o cliente em um, ambiente 'logado', por exemplo.

SR. RUBENS KUHL: A próxima pergunta é do Dario Santana, a pergunta é de como é que você tem visto o PIX, você tem visto muito acesso de momento cadastrado? A gente já deu no chat a URL dos dados do Banco Central, que são dados gerais, aí, de PIX em todas as

instituições financeiras. Mas se você puder complementar com a visão aí da sua instituição.

SR. DIORACI CORRÊA JUNIOR: Certo. Só repete, que cortou o início da pergunta.

SR. RUBENS KUHL: Ele perguntou como tem visto o PIX. Se tem muito acesso de momento cadastrado?

SR. DIORACI CORRÊA JUNIOR: Ah, sim, sim, sim. Cara, é uma crescente bem grande, assim. Muita gente utilizando. O início, quando a gente começou... porque ele teve uma implementação escalonada, não é? No início de novembro, no mês de novembro foi aberto um piloto com até 5% dos clientes, a gente poderia colocar no piloto. E a gente executou esse piloto, algumas pessoas selecionadas para fazer testes de envio de PIX, volta de PIX, e tal. E no último dia 15, 15 ou 16 de novembro, não me lembro, que foi aberto ao público geral. E a gente tem visto um uso crescente, sabe? Ele vem... ele começou com um número X e diariamente a gente vê um crescimento bem grande no uso da ferramenta. Eu acho que até por conta da praticidade dele mesmo, não é? É um produto que muito *friendly*, muito amigável, assim. Você consegue fazer transferência de pagamentos de uma forma bem simples. E isso está incentivando bastante.

E uma coisa que, para mim, foi até surpresa, foi a questão de empresas. A gente pensa muito no PIX como sendo pessoa física, não é? O usuário final, pagando alguma coisa, transferindo dinheiro. Mas empresas também usam e o uso massivo, inclusive, por parte de PJ. E isso é uma coisa que eu não tinha noção que isso poderia ser dessa forma ou acontecer. A gente sempre pensa no mais comum, não é? No usuário final, lá, na pessoa física. Mas, sim, está havendo um crescimento bem expressivo com relação ao uso do produto.

SR. RUBENS KUHL: E a última pergunta que vai poder fazer dentro do tempo que está disponível, o professor Peraro cita que as transações atuais repassam os dados dos titulares junto com a chave para o recebedor. Ele acha que isso é algo que não parece ser interessante. Qual a sua visão a respeito?

SR. DIORACI CORRÊA JUNIOR: Cara, sim, existe essa questão mesmo. A gente se deparou com alguns usuários falando: "Não, eu não quero passar meu CPF, eu não quero passar meu telefone ou o e-mail para alguém para eu transferir dinheiro. Isso é um dado meu, eu não quero passar". Ok. Então, existe até dentro do PIX a questão da chave aleatória, não é? Que é uma chave que não tem vinculação a um dado pessoal seu. É uma chave sua, ela é aleatória, ela é gerada para você e você pode passar e utilizar ela como sendo o seu meio de pagamento, não é? Então, alguém te pede: "Ah, me passa a sua chave PIX", passa a sua chave aleatória que não vai ter nenhuma exposição,

assim, de um dado seu, entendeu? Mas, sim, isso é uma preocupação válida. Exato.

SR. RUBENS KUHL: Dioraci, de perguntas ligadas à segurança da informação, e não o caráter financeiro aí do PIX, é o que a gente tinha mesmo. Obrigado aí pela apresentação. Eu devolvo para o Adriano.

SR. ADRIANO CANSIAN: Obrigado, Dioraci. Obrigado aí por todas as perguntas que recebemos, não é? Só uma colocação aqui do Danton Nunes no chat do YouTube, ele disse que TED é leite Ninho, e o PIX é leite Ninho instantâneo. Você não precisa... Que é mais rápido para fazer. Eu até corrigi ele que é Molico, que a gente não deve ficar comendo muita gordura, sem querer fazer propaganda aí da Nestlé.

SR. DIORACI CORRÊA JUNIOR: É um instantâneo que pode ser tomado de madrugada, inclusive.

SR. ADRIANO CANSIAN: Ah, perfeito. Muito bom. Dioraci, muito obrigado aí pelas colocações. Obrigado a todos. Nós vamos fazer um intervalo agora de 20 minutos, nós retomamos às 10h40, 10h40, para manter o horário, não é? Um pouquinho menos que 20 minutos de intervalo. Então, esse é o GTS 35, evento do Comitê Gestor da Internet no Brasil, organizado pelo NIC.br e pelo Registro.br. Nós retornamos, então, 10h40. Vamos fazer uma pausa aí para a gente fazer um chá, tomar um café, fazer um *home break*. Obrigado, pessoal. Até já.

[intervalo]

SR. ADRIANO CANSIAN: Olá, pessoal. Sejam bem-vindos de volta, com a continuidade das sessões do GTS 35, essa virtual, evento do Comitê Gestor da Internet no Brasil. Relembrando a todos que os slides estão disponíveis via FTP para ftp.registro.br/pub/gts/gts35. É fácil de achar. Se você estiver usando o navegador e quiser trocar protocolo para https, com o mesmo endereço do servidor ftp.registro.br, vai funcionar também. Ainda que nós achamos que você vai gostar de usar um protocolo FTP nativo.

Bom, vamos dar continuidade as sessões dessa manhã, então, convidando o André Grégio, da Universidade Federal do Paraná, e o Marcus Felipe Botacin, também da Universidade Federal do Paraná. Eles vão fazer a apresentação: Integridade, disponibilidade, confidencialidade e ransomware. Bem-vindos.

SR. ANDRÉ GRÉGIO: Obrigado, Adriano. Bom dia a todos. É um prazer estar aqui, depois de tanto tempo, no GTS. Sempre é um prazer estar aqui entre os amigos aí e colegas. Então, hoje a gente vai fazer uma apresentação que é mais uma reflexão, não é? Lógico que tem a

parte divertida, prática ali, que o Marcus vai mostrar, mas é uma reflexão de para onde a gente está indo na área de segurança, que eu acho que é uma eterna reflexão, que não só tem que ser feita, mas a gente vive em um *looping* infinito de desgraças. E parece que nessa corrida de gato e rato, a gente está sempre para trás. E os problemas são sempre os mesmos, não é? Passam-se as décadas e nós continuamos sofrendo dos mesmos problemas, mesmo sabendo dos princípios básicos de segurança e tudo o que tem que ser feito, tudo o que tem que ser construído e implantado e seguido, não é? A tal da conscientização, ela parece que nunca chega. Então, a gente colocou esse título aí.

Então, tem que motivar um pouquinho a palestra, até chegar na parte prática. Então, eu vou mostrar um pouco de um histórico aí do ransomware, como ele evoluiu ao longo do tempo, até chegar nas ameaças atuais, não é? E aí vai ter uma demonstração de alguns exemplares, como eles funcionam, qual é o comportamento deles, até resolvendo alguns mistérios de se é possível recuperar a chave ou não, aí no fim a gente faz um lições aprendidas. Aprendidas, porém constantemente esquecidas, não é?

Todo mundo sabe que um ransomware é um código malicioso que viola a disponibilidade dos arquivos ou dos dispositivos aos quais ele infecta, por meio de criptografia, ou às vezes embaralhamento, ou usa até uma codificação diferente, não é? Ransomware antigo, a gente já viu que faz as coisas sem usar chave forte, né, ou até usando algum método de ofuscação codificando para a pessoa não ver, né, os mais ingênuos, assim. Mas hoje, como tudo em crime cibernético, as coisas viraram uma motivação financeira, organizada por quadrilhas e etc. Tem poucas pessoas querendo aparecer e muita gente lucrando ali com a parte de crime mesmo.

Os tipos principais de ransomware. Existem dois tipos principais, nem todos eles são: ah, cifrou todos os meus dados. Alguns, eles tentam trancam o computador, ele tranca o usuário para fora, são os locker ransomware, esses do tipo *locker*, eles acabam travando a tela e deixa uma mensagem, como se fosse um *screen locker* ali para a pessoa pagar de alguma forma o resgate e depois ter a chave que destrava o computador, né, pode travar alguma partição de boot, alguma coisa do tipo, não é? E os mais comuns, que são os que saem bastante na mídia são esses crypto-ransomware, que eles entram na máquina, de alguma forma, e cifram arquivos, geralmente os documentos, ou eles procuram por tipos de arquivos distintos, por .DOCX, XLS, fotos, etc., e solicitam resgate, em geral, em criptomoeda também, não é?

Como é o comportamento padrão de um ransomware? Porque quando chega e vai discutir o artefato em si, o exemplar que cifrou,

nós estamos falando em um ataque que já ocorreu, não é? É um exemplar de malware que entrou na máquina, de algum jeito, e ali ele fez a atividade dele, não é? Ele fez o comportamento dele. Mas, às vezes, a gente se perde, não é? Até em muitas análises que tem por aí, sempre é em cima do binário, do que ele fez, como é que ele cifrou, como foi a troca de chaves. Mas por que a gente ainda é infectado por código malicioso, né, sendo que a indústria já tem aí quatro décadas de desenvolvimento, e nós estamos tendo um problema básico. Porque não importa como o ransomware faz a cifração ou não, não é? O que importa é que ele está entrando nas máquinas. E isso tem um caminho longo do ataque, não é? Tem toda aquela escadinha de ataque, né, ou aquele procedimento: ó, teve uma varredura, ou alguém mandou um ataque indiscriminado para todo mundo via phishing. Então, não tem só a fase da execução do ransomware. Tem todo um procedimento que começa com a entrega que, em geral, pode ser feita via phishing, não é? É o jeito mais fácil: clique aqui para ver alguma coisa, comprar algum produto, não é? Acesse esse anexo em PDF, seu nome está sujo. E dali vai ter um script, ou alguma coisa que vai permitir, ou mesmo um *downloader*, né, que vai permitir que essa amostra de ransomware entre na máquina.

Mas existem outros jeitos também, né, não só o phishing. Algumas mídias sociais, ferramentas de mídias sociais ou grandes sites já foram alvos de anúncios que permitiam o download de ransomware nos seus usuários, não é? Então, só a navegação já lhe permite pegar um ransomware. E tem casos de *pay-per-install*. A pessoa vai lá e quer instalar um programa que aumenta a performance do computador, não é? Ou limpa o espaço mal ocupado, esse tipo de coisa. E quando baixa esses *pay-per-install*, ele pode causar a exploração de alguma vulnerabilidade ou baixar um programinha ali que, no fim, carrega um ransomware, não é? E tem o *exploit* comum, tipo um *Worm*, não é? Vão varrendo máquinas na rede. Esse é o conjunto de etapas mais comum de um ataque, não é? Faz uma varredura, acha máquinas com algum serviço habilitado para a rede em uma determinada versão, e explora uma vulnerabilidade, seja do Windows, do Java, etc., e se propaga, e quando chega lá dentro da máquina, aí sim, tem o *dropping* ou o *downloading* do ransomware.

Então, a segunda fase seria a execução, que é esse comportamento padrão que a gente já sabe, eu vou procurar um determinado diretório, tipos de arquivo, compartilhamentos em rede, acessá-los ou criptografá-los para depois exibir o aviso de resgate. Solicitar, então, o pagamento, que pode ser que abra o browser na sua cara com um HTML lá, ou pode ser que trave a tela, pedindo alguns bitcoins para uma ou mais carteiras virtuais.

Existem alguns artigos. Assim, vocês vão ver ao longo da apresentação, tem vários links, que são materiais complementares

para quem quer entender o *underground economy* por trás do ransomware, como é que funcionam algumas defesas, pelo menos acadêmicas, né, depois a gente discute isso. Mas existem várias formas do atacante, depois, sacar esse dinheiro. E isso é tema para um outro dia. Porém, o importante aqui da parte do pagamento é que nunca é garantido, né, que a chave ou qualquer outro meio de decifragem necessário seja fornecida após a vítima pagar o resgate. E na maioria dos casos, realmente, não há o pagamento, não é? Existem inúmeros notícias na mídia, até de órgãos públicos que pagaram mais de uma vez resgate e continuaram sem acesso. Porque o atacante também não vai lá ficar esperando, não é? O comércio dele não é vender a chave, e sim receber o dinheiro e mandar o ransomware para a frente.

Já a decifragem, depois desse pagamento, supondo, no mundo ideal, onde você paga o resgate e o sequestrador te entrega a vítima, pode ser feito por um binário, que depois de confirmado esse pagamento, lá, ele recebe na carteira virtual dele, do cliente - é que o crime é bem organizado mesmo - vem um binário para ser baixado ali, ou alguma outra forma de decifrar.

Os dados. Eu peguei uma estatística aqui da McAfee, do relatório mais atual, agora de novembro, e mostra que... a quantidade de novos ransomwares encontrados, embora tenha diminuído ou se estabilizado, né, do último trimestre de 2019 até os seis primeiros meses desse ano, nós estamos com mais de 1 milhão de novos ransomware identificados, não é? E cada um deles, né, tem aqui todo o nome das famílias, ao longo do tempo, eu tirei desse artigo, eles têm de(F) algumas dezenas de milhares de dólares até alguns milhões, não é? Se a gente pegar famosos, como o WannaCry, ele não teve um prejuízo para as vítimas muito grande, né, pela extensão do ataque. Mas mostrou como um dano global pode ser feito. Agora uns mais famosos de usuários, né, que usuários pegaram, como o CryptoLocker ou o CryptoWall, tem aqui uma taxa de resgate pago, né, um prejuízo bem grande, US$ 3 milhões, e US$ 18 milhões, o CryptoWall. Então, é bastante grana aí a ser somada.

Mas quando a gente chega, assim, em como detectar essas ameaças, não é? Tirando essa parte do processo de ataque, se você tiver uma rede fechada, com uma topologia bem segregada e etc., né, como é que a gente chega na parte de: ó, teve um programa que fez download, eu quero saber se ele é um ransomware ou não. É a grande questão em cima da pesquisa em códigos maliciosos. Porque a diferença de um programa malicioso ou não é a intenção do que ele faz, não é? A intenção do criador dele para as atividades que ele vai fazer no sistema infectado. Porque o resto é tudo linha de código, igual qualquer outro programa, não é? Um bot e um *instant messenger*, eles podem ter as mesmas funcionalidades, a detecção acaba se tornando muito difícil.

Aí quando chega na parte da indústria, tem muitas promessas e, na prática, é muito difícil se cumprir essas promessas, não é? As coisas vão mudando de nome, mas o interior delas continua o mesmo, não é? No começo lá da era dos antivírus, ali, no início da década de 80, o que se tinha? Eram vacinas específicas, não é? Alguém lançava um vírus, alguém codificava uma expressão regular que detectava se tinha o padrão no cabeçalho dos arquivos para tentar remover aquele vírus. Depois virou o antivírus, né, que já tinha um conjunto de expressões regulares. Daí foi para antimalware, porque não tem mais só vírus de computador, definiu-se um monte de classes, além do Worm, Trojan, Backdoor, tem uma miríade de classes. E hoje chama anti-APT, e essas *advanced persistent threats*, na verdade, acho que elas são o menor dos problemas. Porque os vírus mais ordinários, sem vergonha e mal codificados do planeta continuam acessando o usuário. Então, se resolve o problema básico, e quer se resolver um problema avançado, não é? Sendo que, na prática, a gente sabe que quando você quer pegar sinais ou indicadores de compromisso que são muito esparsos ao longo do tempo e muito disfarçados, muito dificilmente vai se detectar uma ameaça persistente avançada. Talvez anos depois que seus dados já foram todos roubados, detecte-se e se fale: olha aí, funcionou o meu sistema, aqui, inteligente, 'machine learneresco' e artificial aqui.

Então, na teoria, a detecção acadêmica, quando a gente pega os artigos, também a coisa não é muito melhor, não é? Muitas das coisas são feitas off-line: ó, peguei esse conjunto de malware aqui de mil novecentos e bolinha e... que legal. Tem um algoritmo que detecta 99.9%. Na prática, tem que se ficar ou fazendo *scanning*, que tem um custo de complexidade muito alto, ou aplicando classificadores que podem sofrer de desvios no treinamento, muito grandes, porque eles só funcionam para aquele conjunto que foi treinado e eles não se abrangem para novos exemplares, não é? Ou eles... ou experimenta, geralmente, o software, não é? E na prática a gente tem aquela solução de mercado que está tentando pegar a coisa mais ordinária e colocando um nível de sofisticação em cima. Mas com a limitação, também, que você não pode monitorar todo e qualquer processo e todas as chamadas de API, chamadas de sistema de todos os processos na máquina do usuário, senão pode acabar comprometendo algum núcleo de processamento ou tornando a máquina indisponível. Se tudo a gente colocar em quarentena antes de deixar o usuário usar. Ó, deixa eu ver esse programa executando um pouco para ver o comportamento dele, se é malicioso ou não, o usuário não tem paciência de esperar, não é? Se a gente considerar que hoje, também, todo mundo usa dispositivos móveis e com bateria, você não vai ficar lá escaneando N processos, batendo N assinaturas e tentando, também, fazer uma cadeia de chamada de sistemas que represente um comportamento

malicioso se não acabou a bateria antes... e fritou, né, o celular da pessoa.

Então, há décadas a gente sabe os princípios básicos, que é proteger a integridade, disponibilidade e confidencialidade. Realizar primeiro a prevenção, colocando firewall, colocando políticas de segurança, colocando detectores de intrusão, bota honeypot para estudar. Implanta as políticas de verdade, com apoio das chefias e direções, mantenha tudo atualizado da máquina. Compartimentalize.

Mas por que isso não tem funcionado? Então, a gente tem alguns problemas, que eu acho que é entre essa detecção acadêmica e a detecção industrial, que ambas, elas têm as suas limitações, mas a gente tem um problema lá, mais velho de todos, da humanidade, que é o próprio ser humano. O ser humano não só estraga o planeta, ele estraga o uso computacional, não é? Se tem um air-gap, né, um malware que está dentro de um pen-drive ou de alguma mídia que tem que ser inserida no computador para que ele fazer o ataque, pode ter certeza que esse malware vai encontrar um jeito de entrar na rede.

Então, aqui tem uma linha do tempo de alguns códigos maliciosos, desde 89, que representam ransomware. E o primeiro deles, que é bastante interessante, era esse PC Cyborg aqui. Que um biólogo, o Dr. Popps, espalhou em uma conferência alguns disquetes com um questionário sobre Aids, tinha aqui mesmo um questionário, né, então, esse ransomware também era um Trojan, porque a pessoa podia responder o questionário dela, mas ele tinha um contador com um instalador ali. Então, ele infectava o AUTOEXEC.BAT ali, e depois de 90 *reboots*, aí ele criptografa, criptografava simetricamente os nomes dos arquivos, não é? Porque é importante criptografar os nomes. Se o computador for executar um arquivo baseado na extensão, apenas, e não no filetype, ele não consegue fazer com que esse arquivo funcione novamente, que ele seja aberto, não é? Tem muitos programas que leem a extensão para poder fazer isso.

E essa criptografia simétrica dos nomes, se você não sabe mais o que é extensão o que é arquivo, às vezes pode se tornar difícil, se tiver centenas de milhares de arquivos, para depois fazer a decifragem. Porém, como era simétrico, foi fácil criar uma vacina que deduzia a chave. Então, tem um artiguinho na Virus Bulletin que conta toda a história do aqui PC Cyborg, também chamado como Aids, que é bastante interessante de se ler. Então, foi considerado o primeiro ransomware, porque ele pedia, se vocês virem aqui na tela vermelha, para pagar US$ 189 por uma conta lá no Panamá, não é? Era o bitcoin da época, né, a carteira virtual privada da época.

Então, tem aqui algumas análises. Mas aí, em 1996 tem um artigo da *Security and Privacy.* O primeiro vírus de computador a ser definido formalmente foi na tese do Fred Cohen, em 1983, foi o

doutorado dele em Stanford. E aqui a gente vê mais uma inovação acadêmica, né, na *Security and Privacy* de 96, esses dois pesquisadores, o Yong e o Yung, eles mostraram a implementação de um vírus com criptografia de chaves públicas, que eles chamaram de *cryptovirus*. Então, isso aqui foi a base de ransomware que a gente viu a partir dos anos 2000. Então, teve lições aprendidas depois Aids e do *cryptovirus* para todo mundo, não é? Para os desenvolvedores de malware, é não usar a criptografia simétrica, porque era fácil, depois, de os analistas quebrarem e restaurarem os arquivos, aí eles não ganhariam o resgate. E com o surgimento do *cryptovirus*, os usuários e organizações deveriam ter aprendido uma lição, que é que o backup é importante, não é?

Nós tivemos, depois aí dos anos 2000 para a frente, diversos tipos de artefatos cujos comportamentos são mais ou menos padrão, né, vem por e-mail em um phishing, o usuário executa o artefato, ele vai lá e cifra, não é? Alguns são variantes de um grupo, do mesmo grupo, outros têm o mesmo comportamento, mas a base de código é diferente. O que indica diferentes grupos, mas a gente sabe que a atribuição de criminosos é um problema em aberto também.

Então, tem alguns casos aqui para citar. Que é mais um histórico, não é? Tipos de arquivos que os ransomwares pegam, como o CryptoWall funcionava, já era um pouco mais sofisticado que o CryptoLocker, não é? Além de vir por phishing, tinha alguns *exploit kits* ou propagandas maliciosas que serviam de vetor de entrada. Mas, no fundo, ia lá tentar cifrar e cobrar o resgate.

Depois, uma outra inovação foi o ransomware *as a service*, onde os atacantes começaram a vender kits para qualquer um poder soltar o seu ransomware na prática. E aí esses atacantes, eles ganhavam em comissão. Se você der uma comissão ali do resgate, leva o fazedor de código, não é? Então, tem algumas estatísticas aqui, de 5 a 30% de comissão para usar ransomware alheio. A coisa foi se especializando cada vez mais. O Chimera, a inovação dele foi usar o *doxing* aqui, que os autores prometeram disponibilizar os documentos das pessoas, caso o resgate não fosse pago. O que é uma boa, do ponto de vista da vítima, a menos que ela tenha documentos comprometedores, não é? Se o criminoso disponibiliza o documento, ela pega de volta já decifrado, então, não precisa pagar o resgate, né, é mais fácil. Chantageia aí, põe meus documentos on-line que eu baixo eles de volta. Acho que esse foi o grande erro do Chimera ali, não é?

E as tendências de 2020, na área de ransomware. Todo mundo está trabalhando remoto, praticamente, tem que acessar a empresa, muitas organizações afrouxaram as suas políticas de acesso remoto. O RDP está habilitado em todo lugar. Já estava, não é? Não dá para dizer que foi hoje, com o trabalho remoto, que abriram o RDP. Mas a gente

vê alguns exemplares mais novos cuja entrada, uma das formas de entrada é por força bruta de RDP, daí você tem que ter uma política boa de senhas e com usuários conscientes, também, né, na hora de fazer os acessos. Então, tem várias coisas que não dá para controlar sendo um analista de segurança. Porque tudo o que a gente fizer, vai ter o usuário ali para colocar a senha dele, para plugar o computador dele para deixar o filho usar o computador e pesquisar sites, e clicar em anúncio e etc., não é?

E aí chega no mês passado. Então, teve um problema de ransomware, alguns links aqui de um caso famoso. E o que a gente vai ver, eu coloquei o link para algumas análises na Corvus, o nosso sistema de análise dinâmica. Aqui não mostra muita coisa, afinal, são códigos bem simples, mas simples não significa simplista, do ponto de vista ignorância, não é? Simplicidade, às vezes, é elegante. Eu vou deixar o Marcus mostrar a parte dele. Nós recebemos dois exemplares, um PE, que é um *loader*, e um ELF com execução manual. Esses artefatos foram os mesmos... estiveram envolvidos nos ataques do Texas *Department of Transportation* aí em maio. Então, gente, como sempre, foi atacada até depois. Marcus, eu vou tirar para você...

SR. MARCUS FELIPE BOTACIN: Tá bom. Todo mundo consegue me ouvir?

SR. ANDRÉ GRÉGIO: Sim, senhor.

SR. MARCUS FELIPE BOTACIN: Então, tá bom. Então, no final eu apareço, eu vou dar *share* na tela cheia aqui, para todo mundo conseguir ver. Bom, então, eu espero que todo mundo esteja conseguindo ver agora.

Eu vou falar um pouquinho de ransomware, tá? Sobretudo do ponto de vista do analista. Então, caso alguém nunca tenha visto um, eu vou só mostrar que ransomware é esse tipo de cara aqui, ó, em que você tem um arquivo na sua máquina, que é do seu interesse. E quando você roda o ransomware, depois da execução dele, o seu arquivo já não é mais o arquivo original. Ele é o arquivo modificado pelo atacante, criptografado, codificado, e assim por diante, não é?

Então, do ponto de vista do analista, o que as pessoas sempre perguntam é: Tá, e agora, dá pra reverter? Bom, a melhor política para reverter é sempre o backup, que no meu caso é simplesmente voltar o comando aqui. Mas, se não é o seu caso, se você não tem o backup, vamos olhar o que o analista pensa um pouquinho. Então, eu vou abrir aqui no *debugger*. E bom, quando você fala em reverter o ransomware, no fim das contas, você quer achar a chave. Então, sempre, o analista, ele vai procurar é a função que toma conta da chave. Então, nesse caso, é essa função aqui. Eu vou tentar passar rapidinho para não tomar muito tempo. Então, quando a gente chega nesse ponto aqui, a gente vê que aqui tem uma função, faz(F) uma série de coisas, não

vou ficar detalhando muito, por enquanto. E que uma dessas coisas que essa função faz é obter a chave.

Então, quando a gente chega aqui no final, nessa última instrução, seja lá o que for que essa função faz, mas quando eu chego aqui, eu sei que essa chave está em memória e ela vai ser retornada para o programa, certo? Então, se eu tiver acesso a esse ponto, eu consigo saber onde está a chave. Então, nesse caso aqui, o próprio *debugger* está me falando: aqui é uma variável global que está nesse endereço. Então, se eu for até esse endereço, e falar: imprime para mim o que está nesse endereço. E ele vai falar: legal, aqui tem esse valor aqui, certo? Que é esse valor aqui da chave.

Bacana. Então, eu posso deixar o ransomware rodar, e mesmo depois que ele terminar, que eu tiver o meu arquivo aqui já comprometido, se eu rodar a minha função de recover com a chave que eu identifiquei, o arquivo volta a ser o arquivo que era antes. Pelo menos, no mundo ideal. Claro, se eu identificar a chave incorretamente, o arquivo é corrompido, é escrito lixo ali, porque não era essa chave, não era isso que ele vai decodificar.

Então, parece um cenário legal, não é? Ransomware não tem nada de mais. Em algum momento a chave vai estar ali, vai ser escrita. Mas bom, tem um probleminha. Então, eu vou botar o *breakpoint* aqui de novo e vou rodar outra vez. E a gente vai perguntar de novo qual é a chave. E a gente vê que agora a chave é diferente. O será que está acontecendo? Bom, eu vou rodar de novo. Quando chega aqui e pergunta qual a chave, a chave é diferente de novo. Por quê? Porque essa chave é gerada aleatoriamente. Então, agora ficou mais difícil. Porque agora eu não estou mais rodando em um *debugger*, eu tenho uma infecção que já ocorreu, esse malware já rodou, eu sequer sabia que ele rodou, então, eu não monitorei, e ele gera uma chave aleatória. Então, se eu inspecionar ele posteriormente, eu já não sei qual é a chave mais. Bom, tem problema.

Dá para recuperar? Aí volta a mesma pergunta para o analista. E a resposta é: depende. Então, vamos dar uma olhada, de novo, na função get_key. A gente vê que em um determinado momento aqui, ele chama função rand. Então, essa função rand é que está gerando essa chave aleatória. Você só vai conseguir reverter se tiver alguma falha aqui. Nesse caso, tem uma falha já clássica, não é? Por quê? Porque a função rand, ela não é aleatória de verdade. Ela é pseudoaleatória, ela precisa de uma semente, né, que você dá com srand, e a semente passada está aqui, ó, *time*, né, mov(F) aqui em cima, mov(F) no zero. Ou seja, a hora local. Então, se você, de algum modo, através de um mecanismo de log, ou através do [ininteligível] no FileSystem souber a hora que esse ransomware rodou, você, em tese, consegue colocar o valor de *time* aqui, de volta, né, 'resetar' esse

horário para a hora da infecção e gerar a mesma chave. Por quê? Porque está vendo aqui em eax? eax vai conter o retorno da função *time*. Ou seja, vai ser a hora que vai ser retornado para o *time*. Então, se eu fizer *break* aqui e olhar nesse ponto o eax, a gente tem essa semente aqui. Esse valor aqui, que é utilizado como semente para a função, que gera essa chave aqui.

De novo, se eu rodar outra vez, a gente vai ter outro valor em eax, vocês podem ver que agora o valor é diferente, consequentemente, a chave é diferente. Mas, como eu falei, se eu chegasse nesse ponto e falasse: não, o meu eax, agora, vai ter aquele valor que eu salvei. Eu descobri que foi essa hora que o malware rodou, posso colocar aqui, continuo a execução, e quando eu perguntar qual é a chave? xddj. Podem voltar lá em cima, olha lá, xddj. Ou seja, sempre que eu rodar com esse horário, ele gera essa mesma chave. Então, se ele tiver uma falha de implementação, eu vou conseguir reverter.

Bacana, não é? Mas esse é um malware de exemplo que eu selecionei. Agora, vamos rodar um malware de verdade. Então, esse é o ransomware que rodou nesse caso do STJ, pelo menos é a mesma família, não é? E para rodar, é(F) todo um argumento, o que significa que a infecção já tinha ocorrido. O cara já tinha controle do sistema quando ele rodou. Eu vou passar o ponto aqui mesmo e vou rodar. Legal. Agora, quando eu olho os meus arquivos, vocês podem ver, mudou, não é? Mudou até a extensão dele. Se eu olhar para o myfile, vocês podem ver, completamente encriptado. E além de tudo, eles ainda deixam essa *news* aqui, que é a ransomware *note*, que é onde ele avisa você que ele comprometeu o seu sistema, enfim, seus dados foram criptografados, faz um pouco de terror aqui. E em um certo ponto, ele fala: olha só, você pode enviar os seus arquivos aqui para mim, por e-mail, que eu vou decriptar eles para você. Guardem essa informação que ele vai ser muito importante daqui a pouco.

Eu vou sair aqui da máquina virtual e vou passar para a análise dele aqui no IDA para mostrar um negócio. Então, eu não vou entrar em tantos detalhes, mas eu quero dizer que, bom, temos aqui a *main* do programa, em certo ponto ele chama aqui, ó: GeneratePreData. Basicamente, a gente pode imaginar o que ele vai gerar aqui, não é? Ele vai gerar a chave para posteriormente encriptar. Quando a gente vai em GeneratePreData*,* finjam surpresa, olha o que encontra: *time,* srand em rand, certo? Então, o ransomware de verdade não acaba sendo muito diferente daquele ransomware didático que eu apresentei. Não tem grandes diferenças.

Mas não adianta ficar feliz agora. A gente poderia pensar: 'Pô', legal, é o mesmo problema, então, se a gente souber o horário, a vai conseguir reverter, não é? Bom, nesse caso, o atacante, ele é um

pouquinho mais esperto, né, do que um atacante didático, ele tem uma implementação real e a gente vai ver aqui que ele usa entropia, não é? Então, ele inicia uma entropia aqui, em vários pontos, aqui em cima e aqui embaixo, não é? E aqui ele vai chamar o número aleatório, aqui, ó, debug(F) random. Então, além de gerar a hora, a rand com a hora local, ele vai somar isso com uma entropia que é gerada localmente, e aí sim, essa chave fica verdadeiramente aleatória e é por isso que a gente não consegue reverter a criptografia desse ransomware.

Isso gera uma outra pergunta interessante, não é? Bom, se é totalmente gerado aleatório e local, como é que o atacante sabe qual é a chave que foi utilizada para encriptar esses dados? E a resposta, mais legal ainda, é: ele não sabe. Por isso que a gente vê aqui que ele tem um rsa_init, aqui, ó, nesse ponto, onde eu estou com o mouse. Por quê? Porque quando ele gera a chave, simétrica, que vai criptografar os dados, ele também cripta com a chave pública dele, assimetricamente, essa chave, apenda no arquivo criptografado, e como a gente viu lá na mensagem de ransomware, ele pede para você enviar o arquivo para ele. Quando você envia o arquivo para ele, o que ele faz, na verdade, é primeiro, abrir, com a chave privada dele, o arquivo, para descobrir qual a chave simétrica que encripta o restante daquele arquivo. Então, nem mesmo o atacante sabe qual é a chave que ele utilizou, certo?

Legal, esse foi o primeiro caso, foi o caso do ransomware do STJ. Mas esse não é o único. Há outras perguntas que a gente precisa se fazer. Por exemplo, como é que acontece no Windows? Porque as pessoas continuam caindo nisso, não é? Então, eu queria esse outro exemplo, que é esse VBScript, né, um script do Windows, chamado Coronavírus. Também apareceu em um órgão governamental, assim que a pandemia começou. Se a gente olha o conteúdo dele, basicamente o que a gente tem é isso aqui, claramente inteligível para um ser humano e eu diria para um antivírus. Porque a gente não tem nenhuma chamada de função aqui, não tem nada em claro. O antivírus teria que interpretar tudo isso para entender o que está acontecendo aqui. Então, provavelmente ia acabar passando, certo?

Bom, isso aqui basicamente é ofuscação, não é? Então, se a gente clarear um pouquinho esse script. Eu já tive esse trabalho, o que a gente vai ver é que o que na verdade a gente tem é um código aqui em cima, que para quem está acostumado com ofuscação, vai ver que tem uma cara(F) de Base64, isso aqui, tem alguns arrays aqui, né, vários [ininteligível] que para quem está acostumado com ofuscação, vai imaginar que isso aqui seja [ininteligível], ou seja, no fundo, esses [ininteligível] aqui vão ser manipulados para virar [ininteligível] e em tempo [interrupção no áudio] esse *array* vai virar um comando, né? Como a gente pode ver aqui, ele está decriptando isso, desofuscando, decodificando. Tem uma variável de [ininteligível] gigantesco aqui,

mas é só para te enganar. No fundo é uma variável qualquer. E no final executar isso aqui, que provavelmente é o comando que ele gerou. Vamos tentar clarear mais um pouquinho. Tinha *payload* lá. Na verdade, esses dois *array* a gente vai ver que eles vão ser percorridos inteiro, ele vai fazer essa conta da raiz quadrada de cada um deles e ele vai executar um comando que vai apendando caractere por caractere a diferença daqueles dois *arrays*. Então ele vai gerar um [ininteligível] e apendando letrinha por letrinha do comando e no final vai executar o comando. Então o usuário teria que conseguir identificar tudo isso ou antivírus teria que seguir todo esse fluxo para identificar que isso está acontecendo e qual é o comando gerado. Eu já segui esse fluxo antes, então eu vou mostrar o que é aquele comando que ele gera. E aquele comando que ele gera é outro script. Então, na verdade, ele é um malware modular. Ele vai etapa por etapa gerando outro *payload.* O que vai ter nesse *payload*? Aqui ó, decodificação de base 64, porque o conteúdo malicioso está dentro dele, então ele precisa extrair dentro dele. Extrai para aonde? Extrai um diretório temporário num arquivo EXE, ou seja, além de multiestágio, esse malware, ele é multiformato. Porque antes a gente lidando com script, agora a gente está lidando com binário típico. Então o antivírus teria que parsear(F) todos os diferentes tipos de formato que esse cara utilizou para conseguir pegar ele. Bom, e quando a gente roda esse cara? Não tenho muito tempo para mostrar toda execução dele, mas depois que você roda, adivinha o que acontece? [ininteligível], de novo seus arquivos criptografados, pedindo resgate, geração de chave aleatória, basicamente da mesma forma, tanto no Windows quanto no Linux, certo? E por que eu estou falando tanto de malware multiestágio? Porque isso é o que faz na prática um by pass(F) no antivírus. Então queria mostrar uns outros casos aqui de by pass(F). Eu vou tentar restaurar esses [ininteligível] aqui, fazer live é sempre problemático. Pedir um pouquinho de paciência para vocês.

Então, legal, esse é um exemplo que eu vou mostrar como by passar um [ininteligível] real usando simplesmente essa separação dos payloads. Então aqui eu tenho um payload malicioso, em claro, que se eu colo aqui na área de trabalho, tenho antivírus rodando, vocês vão ver que rapidamente o próprio antivírus detecta essa ameaça, porque ele sabe parsear(F) esse tipo de formato e ele remove essa ameaça. Então a ameaça vai ser removida, certo? Vai ser bloqueada e depois ele vai ser removida. Contudo, se eu codificar essa ameaça igual o atacante faz, eu vou colocar esse outro arquivo aqui, vamos ver se vai funcionar. Pronto, é o mesmo nome, só que agora vocês veem que o Windows não reconhece mais como DLL, porque está codificado, ele não está mais apresentando a estrutura de um arquivo executado tradicional. Mas eu tenho o loader aqui que sabe abrir esse formato, então em memória ele vai abrir esse conteúdo e vai executar, certo? Então quando executo ele, a gente pode ver, a gente roda. Ele exibe a

mensagenzinha que eu falhei no clique duplo aqui, e tudo bem. Executou, o antivírus não pegou porque o antivírus escaneia o arquivo, não escaneia a memória e acaba passando, se fosse um conteúdo malicioso, teria passado. Por que eu dei esse exemplo? Não é para mostrar, olha, como o cara sabe by passar [ininteligível], não é nada disso. É porque é uma técnica razoavelmente simples, mas que torna muito efetivo o ataque, porque é difícil para o antivírus correlacionar fontes de dados.

Então eu vou mostrar um outro exemplo, agora de novo um exemplo real, que é um outro payload que afetou o STJ. Eles tiveram um payload em Linux e um payload em Windows. O payload é isso aqui, esse notepad. Se a gente roda ele, o máximo que a gente vai ter são essas mensagens de erros. Então vocês vão ver, vai ter uma série de erros, porque ele não encontra os componentes, no final ele vai 'crashar' e não vai executar, por quê? Porque como falei, é um loader, ele precisa de um outro arquivo para executar. Então já tinha feito análise dele aqui, se a gente procurar aqui, a gente vai ver que tem um arquivo config.dat que ele está tentando carregar e ele não encontra. Tem um name not found ali, ele não achou esse arquivo, por isso que ele 'crasha'. Então na verdade esse ataque foi multiestágio, o atacante colocou mais um componente ali, um [ininteligível] e o payload malicioso para executar. Ele tinha total controle desse ambiente. A gente não tem payload para mostrar aqui, eu não sei se esse payload se perdeu, se simplesmente não forneceram. Não sei. Mas então eu escrevi o meu próprio payload para demonstrar. Porque o payload é algo que parece com isso daqui, né? É um shellcode, você vai escrever as instruções ali [ininteligível], vai codificar e vai fazer ele rodar. Então eu vou copiar aqui para o local que ele está esperando, né? E quando eu rodar esse exemplar a gente tem agora aqui a mensagem GTSGTR(F), que é o que eu tinha codificado para ele escrever. Ou seja, ele carregou o conteúdo do meu arquivo externo e está executando isso a partir da memória. Quando a gente divide os payloads fica mais difícil ser detectado e provavelmente a infecção aconteceu dessa forma.

Vou voltar para o André finalizar, que eu acho que a gente está no limite do tempo já.

SR. ANDRÉ GRÉGIO: Estou recompartilhando a tela aqui. Bom, então como vocês viram aí, o Marcus mostrou todos esses exemplos rodando, diferentes tipos de códigos maliciosos e que são de programação simples, né? O funcionamento deles é simples. E a recuperação pode ser simples ou pode não ser. Teve ransomware no começo ali, acho que [ininteligível] era 1.0, onde eles cifravam o conteúdo dos arquivos e não o cabeçalho. Então para o usuário, ele podia ver: ó, esse aqui é meu documento, sei lá, relatório.doc. E por ter um cabeçalho, por eles fazerem uma codificação fraca, se eu não

me engano era um [ininteligível], era possível encontrar a chave usada no [ininteligível] e decifrar esses arquivos. Mas a tecnologia para se fazer ransomware com infraestrutura de chaves públicas está ali desde o artigo do [ininteligível] de 96.

Então para aonde que a gente vai Educação ainda é a chave, porque não tem outro jeito de resolver. O atacante sempre vai tentar colocar uma [ininteligível] para by passar antivírus, mudar o padrão, fazer alguma firula, colocar N módulos que tentem dissimular o download do artefato malicioso em si, pode fazer em partes e montar depois. E a única forma de proteção, tirando toda a segurança em camadas, é que não se pode negligenciar o backup. Acho que a gente acaba negligenciando muito o backup, eu coloquei cartilha a Cert aqui, usei o folheto deles e coloquei de referência. Porque uma coisa que eu já vi na prática aqui até na universidade é: você pode ter o backup diário, mas se não tiver a política correta para testar o backup, na hora de fazer o [ininteligível], ele não vai funcionar. Não adianta nada ter aquele monte de fita ou discos com inúmeros backups incrementais diferenciais se na hora que se precisa ele não está ali.

Além disso, tem todas aquelas dicas de proteção básicas, né? Mas que muitas vezes a gente também negligencia, além de ter o backup, né? A pessoa pode deixar o backup na rede direto, isso se o ransomware infectar e achar o backup na rede, vai cifrar o backup, mas usar bloqueador de Ads, porque muitas vezes vai enchendo o saco colocar muita ferramenta de segurança porque elas ficam apitando tudo ao mesmo tempo. Como a gente é muito orientado à usabilidade, em geral, tem muitas funcionalidades desnecessárias por padrão que vão permitir a execução de um código JavaScript. Aquele VDS(F) que o Marcus mostrou, existem códigos de JavaScript que são similares no sentido da ofuscação do script, que vão pegando [ininteligível] e montando para depois avariar e fazer a execução. Então, são coisas óbvias, mas que se todo mundo tivesse seu backup, bloqueador de Ad e tudo aqui, o ransomware não seria tão efetivo, né? Você poderia simplesmente: Perdi dado de hoje, vou restaurar o de ontem.

As considerações finais são: faça backup, certifique-se que o backup restaure, proteja os backups de rede. Não adianta ter um backup de rede que ele vai ser infectado também e ter backup off-line. Isso são formas de proteção que são mais antigas do que andar para trás, né? E a gente tem que levar isso em conta. Então a educação ainda é a chave. Todo mundo tem de lição de casa ler o fascículo de backup do Cert e refletir assim "estou fazendo meu backup corretamente?" Acho que era isso que a gente tinha para mostrar. Alguns links de referência de leituras interessantes aqui no final.

SR. ADRIANO CANSIAN: Muito bom, André, muito bom mesmo. Parabéns, André, Tiago...

SR. ANDRÉ GRÉGIO: Marcus, Marcus.

SR. ADRIANO CANSIAN: Marcus, desculpa. Marcus Botacin, foi bem legal a demonstração também. Rubens, temos algumas perguntas aí? Temos pouquinho tempo. Selecione para a gente.

SR. RUBENS KUHL: Temos sim. Então eu vou agregar três perguntas que se referem a sistemas operacionais. Então perguntando se existe um ranking de sistemas operacionais mais suscetíveis. Outra pergunta [ininteligível] se são mais efetivos em Linux ou Windows. E também se sistema [ininteligível] como podem ser infectados e como se proteger desse tipo de ataque. Então a pergunta que vai dar tempo de fazer é essa, com o vetor de sistemas operacionais.

SR. ANDRÉ GRÉGIO: Beleza, bom, Marcus, eu vou responder essa, tá?

SR. MARCUS FELIPE BOTACIN: Pode ir.

SR. ANDRÉ GRÉGIO: [ininteligível]. Assim, não tem esses rankings, eu não gosto de falar dessas coisas, de culpar um sistema operacional porque o sistema operacional, o dispositivo, a aplicação que é mais popular, obviamente, ela é mais propensa a ser alvo de um ataque. E as outras perguntas. A própria demonstração ali do Marcus mostrou, o artefato era um artefato de Linux. E as redes hoje são muito heterogêneas, a gente tem máquinas Linux, tem máquinas Windows, todas elas estão rodando serviço, todo computador que está na rede rodando um serviço é vulnerável, é só procurar no [ininteligível] lá, que tem [ininteligível] para tudo, qualquer versão, né? Se você está rodando um WordPress vulnerável, não importa se você está rodando no Linux, no Windows, no Mac iOS, vulnerabilidade é vulnerabilidade.

Bom, é isso. Tudo é vulnerável, desde que você não tenha salvaguardas de segurança necessárias. Uma rede, não adianta nada ela ser superfechada, ter OpenBSD(F) no servidor de e-mail dela e etc. se por dentro o usuário leva o celular dele, o tablet, o computador compartilhado e se conecta na rede interna e não tem nenhuma segregação de rede de guest. Então se ele estiver com a máquina dele infectada, ele vai varrer a rede interna e vai ser ele o vetor de ataque. Então é bem complicado. Por isso que a educação ainda é a chave, seguida da implantação correta das políticas para [ininteligível] e fechar todas as coisas, ter redes isoladas e por aí.

SR. RUBENS KUHL: Uhum. O Danton Nunes fez um comentário sobre o papel de sistemas de arquivos de logs, como FS(F), BTRS, e a gente não vai ter tempo de explorar essa questão, mas eu queria só remeter para a apresentação do próprio Danton Nunes numa edição anterior do GTS, onde ele exatamente explora essa possibilidade. De você usar sistemas de arquivo com capacidade de direcionamento para

recuperação aí de um episódio de ransomware. Então, obrigado aos dois. Devolvo para o Adriano.

SR. ADRIANO CANSIAN: Obrigado novamente, André, obrigado, Marcus. Desculpa, Marcus, ter trocado seu nome aquela hora, muito nome na cabeça aqui.

SR. MARCUS FELIPE BOTACIN: Tranquilo, valeu.

SR. ADRIANO CANSIAN: Então esperamos contar outras vezes com vocês no GTS. Acho que foi uma apresentação muito didática, muito interessante mesmo. Muito obrigado.

Vamos passar imediatamente a convidar o nosso próximo apresentador, que é o Manoel Domingues Júnior, do Nubank. O Manoel já esteve outras vezes com a gente no GTS e ele vai falar sobre automações para proteção de aplicações em ambiente Linux. Manoel.

SR. MANOEL DOMINGUES JÚNIOR: Oi, pessoal, tudo bem? Sou Manoel. A ideia é justamente falar de automações, como o Adriano falou. Para a gente começar [ininteligível] importante da apresentação, essa apresentação é minha opinião, né? Ela faz parte do meu trabalho, do mestrado, não reflete a opinião do meu empregador. Isso aqui é mais uma técnica, mais uma camada de segurança que a gente pode adicionar aos nossos sistemas. Então isso aqui não é uma bala de prata, não vai resolver todos os problemas que os processos podem ter, aplicações em ambiente Linux. É trabalho em desenvolvimento, então os comentários que vocês fizerem no Youtube são muito importantes para continuidade dele. Aqui tem uma pequena agenda, só para vocês terem um guia de por onde a gente vai passar.

E, bom, para a gente começar a apresentação e poder criar uma motivação, a gente começa a contextualização falando um pouco de software. Tem uma pequena definição aí dentro, e ele é o ponto central da discussão que a gente vai ter na apresentação. Claro que a gente está falando do ambiente Linux, está restringindo para esse ambiente, mas vocês vão ver, vocês verão que alguns paralelos podem ser feitos com outros sistemas. Um ponto aqui referente [ininteligível] destacar é que a gente quer proteger de incidente de segurança e se preparar para quando eles forem acontecer. Isso com esse software. Ou seja, aqui é importante lembrar que uma boa dose de detecção é interessante para a estratégia que a gente vai falar aqui.

Talvez o primeiro ponto que venha a nossa cabeça para proteger um software é utilizar técnicas e ferramentas de forma que o código desse software esteja seguro. Tem vários desafios nesse ponto, nessa camada. Alguns desses desafios já viu em edições passadas do próprio GTS. A gente uma apresentação de [ininteligível] no GTS 30 [ininteligível] GTS 31, [ininteligível] GTS 32. Um ponto bem importante aqui é que um código que está marcado como sem vulnerabilidades

não mostra que ele é seguro, né? A gente até tem uma estatística de uma ferramenta [ininteligível] que grande parte das vulnerabilidades são encontradas em dependências e dependências terceiras, que para encontrar essas vulnerabilidades em dependências terceiras existe um tempo de desenvolvimento, um tempo de pesquisa.

Outro ponto aí é que ter apenas uma frente de proteção e preparação também não é o ideal. É interessante que nossos esforços de segurança invistam em outras camadas, de forma que quando uma camada falhar, a gente possa contar com outras. Então dessa forma uma segunda camada de segurança no nosso software pode ser obtida com a utilização de algumas técnicas de mitigação de exploração a nível da aplicação. Um ponto interessante dessas técnicas é a compatibilidade delas costuma ser ampla e que grande parte delas são implementadas no compilador. Algumas delas são até uma configuração padrão.

Uma terceira camada que está muito ligada a essa segunda é a utilização de técnicas de mitigação de exploração a nível do sistema operacional. Então sistemas operacionais modernos, aí as últimas versões do Linux, já possuem muitas dessas técnicas já por padrão, já está habilitado lá, você não precisa fazer nada. O ponto aqui que é interessante para quem for usar essa camada é que alguns sistemas baseados em Linux, eles têm uma compatibilidade mais ampla com o sistema de legados. E aí algumas dessas medidas de mitigação podem estar desativadas.

Mais uma camada ali, a nossa quarta camada, que é camada que a gente vai falar com um pouco mais de detalhe aqui, é a camada da minimização de privilégios. Privilégios da minha aplicação com o ambiente que ela roda. Acredito que muita gente talvez já tenha ouvido falar de [ininteligível] Linux. Talvez algumas pessoas só tenham ouvido falar para desabilitar ele. Ou [ininteligível] que são módulos de segurança do Linux. Também existe [ininteligível], que a gente vai entrar com mais detalhes daqui a pouco. E até mesmo umas outras técnicas, como uma técnica de [ininteligível], que é utilizada no [ininteligível], no Google cloud e open source você pode usar, se você quiser. Nesse caso aqui a gente está se referindo, só para deixar bem claro, a minimização de privilégios entre o software e o ambiente. Claro, você pode minimizar os privilégios também entre o usuário do software e o software, mas aqui a gente está falando mais entre o ambiente do sistema operacional e a aplicação.

Uma próxima camada que está, que é uma camada de controle de acesso, que aqui eu botei como autenticação, autorização, account, que é onde a gente quer fazer mais perguntas como: será que meu software pode rodar nesse ambiente? Quais [ininteligível] que o meu software está gerando para a auditoria? Será que eu tenho controle

mandatórios de acesso? E a gente já viu no GTS 31 uma apresentação que falou um pouco mais sobre como, quais são os desafios de gerar esses eventos de auditoria e quais são os desafios de ter o controle granular de aplicação, de permissões. Vale lembrar que tem algumas ferramentas que transitam em mais de uma camada, como é o caso do S-Linux, que ele [ininteligível], por exemplo, que atua, tanto nessa camada quanto na camada de minimização de privilégios e em alguns casos de [ininteligível].

Finalmente, a última camada que eu estou colocando aqui é a camada que, bom, se alguma coisa der errada com seu software, você vai precisar de algum mecanismo gerar uma versão nela, alguma versão nova de seu software, sua aplicação e atualizar os usuários que vão sendo corrigidos. Se você estiver dentro de uma empresa, como software do tipo servidor, isso pode parecer simples. Você vai ter um pipeline de desenvolvimento contínuo [ininteligível] contínuo que vai fazer isso para você. A gente falou pouco disso em [ininteligível], numa apresentação que teve no GTS 30. Agora, se o seu software está na mão de um usuário, isso pode ser mais desafiador. Você pode parar para pensar, fiz uma aplicação aqui, sei lá, um navegador web, e agora eu tenho que fornecer uma nova atualização dele e isso é um desafio. E é claro, a gente tem casos de sucesso com a distribuição de novas lições(F), e um exemplo bem legal é o exemplo do Chrome, ele prepara a nova versão para você e você só tem que recarregar ele.

Bom, outro ponto que a gente tem que trabalhar aqui para deixar nosso contexto mais bem trabalhado é que a forma de fazer software mudou ao longo do tempo. E grande parte dessa mudança, ela ocorreu com a ideia de diminuir o custo da manutenção do software, como do exemplo que botei aqui nessa figura. Nesse exemplo a gente tem um software que interage com várias partes diferentes de um ecossistema. Nós podemos avaliar num software desse tipo com aquelas camadas que a gente trabalhou antes. Então a gente pode ver que alguns pontos das camadas têm complexidade diferente. Nós podemos pensar que essas camadas que são administradas pelo compilador e pelo sistema operacional, elas podem exigir um esforço menor. Então botei elas com esse checkzinho do lado. Tem outras camadas que dependem mais do ambiente, como por exemplo uma camada de update, uma camada de verificação de código, de código seguro, de práticas de código seguro, depende do ambiente empresarial onde aquela aplicação está sendo feita. [ininteligível] software fala com o ecossistema muito grande, tem vários domínios, quando a gente fala de minimização de privilégios entre o software e sistema operacional, isso pode ser um pouco desafiador. Isso pode ser mais difícil porque esse software está fazendo muita coisa.

Bom, sendo que nos últimos anos esse paradigma, ele deu espaço para uma especialização maior do software. Ou seja, cada vez

mais a gente ouviu falar de microsserviços e de softwares especialistas. Então nesse cenário a gente tem um ambiente onde o software talvez não necessito de tanto privilégio, já que cada parte dele pode ser especialista em apenas um domínio. Então, com essa especialização a gente pode ter uma minimização de privilégios que não vai ter tanto esforço para ser realizada como software que interage com vários ecossistemas diferentes. Então talvez essa camada ali de minimização de privilégio esteja mais acessível para gente.

Bom, mas para lembrar mais uma vez, a gente está falando aqui de minimização de privilégios entre o software e o ambiente que ele executa, e aqui no caso da apresentação a gente está falando do ambiente Linux. Então o que a gente quer é limitar o quanto o software pode interagir com o sistema operacional, e limitando essa interação a gente vai querer criar mecanismos de detecção e proteção. Por exemplo, para poder ver que, pô, baixei um arquivinho aqui que era um PDF, ele começou a escrever no disco, ele começou a fazer chamadas ao sistema operacional que ele não deveria estar fazendo, a gente pode fazer um link com a apresentação anterior sobre ransomware que a gente teve. Para quem se interessa por essa técnica de minimização de privilégios, de isolamento de aplicações, ela está mapeada no [ininteligível], então é só ir lá e dar uma olhadinha.

Bom, mas finalmente, antes da gente dar uma olhadinha especificamente numa técnica, eu botei um slide com o contexto histórico, mais para vocês entenderem como essas técnicas, como que a implementação delas evoluiu ao longo do tempo. A gente teve um primeiro caso que eu botei aqui, 1998 menos, porque foi a primeira ocorrência que eu encontrei, mas alguns colegas falaram que foi de 97 ou até mesmo antes, da Libsafe, que era uma biblioteca que ela fazia, você botava no seu binário e ela fazia filtragem do siscom. Logo depois a gente teve algumas iniciativas, como AppArmor e o SELinux, que consegue um trabalho semelhante. Logo depois, durante o esforço de [ininteligível] o SELinux [ininteligível] Linux foi criado uma interface, que é o Linux Security Modules. E logo depois vocês podem ver que o SELinux foi [ininteligível]. Em 2003 a gente tem um trabalho bem legal, o [ininteligível] usando o Systrace, uma ferramenta que ele criou para fazer isso, onde você poderia definir políticas e podia até fazer filtragem baseada nos argumentos desse [ininteligível]. Em 2005 a gente teve o Seccomp, que a gente vai falar um pouquinho. Na verdade, a gente vai falar bem mais do Seccomp com filtros, que é o de 2012, Seccomp BPS(F). [ininteligível] Seccomp onde a gente teve essa filtragem de uma maneira mais flexível e a gente começou a ter uma adoção maior desse tipo de camada de segurança. Tem alguns exemplos, o Chrome, o [ininteligível], outros softwares que utilizam isso e tem até sistemas aí operacionais aí, como o caso do Subgraph OS, que usa um pouco de

syscall para poder fazer esse isolamento entre software e sistema operacional.

Bom, então o que a gente vai falar aqui? Falar sobre filtragem de syscall, em resumo. Vamos começar a ver como funciona filtros de BPF, para vocês sentirem ali, e como a gente coloca em dois exemplos? Dois exemplos simples. Aqui nesse slide a gente tem um filtro, esse filtro não é da minha autoria, ele estava pronto, peguei desse link aqui. Ele é muito fácil de entender, ele faz poucas coisas. E a ideia dele é que ele vai bloquear qualquer chamada, qualquer syscall do tipo [ininteligível], que mapeia as informações do sistema operacional. Aqui o que ele vai fazer vai ser retornar o erro. [ininteligível] aqui na linha 6 e na linha 7, na linha 6 onde está syscall que ele está filtrando e na linha 7 é onde está a instrução dizendo que vai retornar o erro de permissão. A gente também tem o código em C(F) aqui, a ideia desse código em C(F) é muito simples. Esse código em C(F) faz duas syscall ali bem nitidamente no código. Vocês podem olhar na linha 5 que eu faço um getuid para pegar o ID do usuário que está executando esse código e na linha 7 eu faço a chamada da syscall [ininteligível] para pegar as informações do sistema operacional. Repara nesse código que se ele falhar no name, eu vou retornar um código de saída 2. Bom, se tudo dar certo, vai retornar zero. Rodando o programa, o que a gente tem de resultado é o que está aqui na tela. A syscall do getuid funcionou, ela não estava sendo filtrada, então ela foi permitida, mas o programa foi encerrado quando a syscall name(F) foi invocado. E aqui vocês podem ver que não foi um encerramento abrupto, ele simplesmente [ininteligível], verificou que ocorreu um erro e retornou 2. Sendo que a gente pode ver que um caso mais elaborado, se a gente quisesse pegar um caso mais elaborado, esse programa BPF que mostrei no início ele pode não ser suficiente, ele está lidando aqui com uma denial list, a gente está alistando o que a gente não quer, não é uma abordagem muito sadia. Então aqui a gente tem um outro exemplo, só para vocês virem a mudança de um para o outro. A única mudança que teve aqui é de ao invés de eu filtrar syscall [ininteligível], eu estou filtrando a syscall getuid, e aqui eu tenho um programa, esse programa, ele só tem uma syscall, a getuid. Essa syscall é um pouco diferente, ela retorna um inteiro. E bom, quando eu executo o programa, vocês podem ver que ele falha, dá erro. Detalhe, como ele retornou erro? Retornando um ID invalido, né?

Bom, usando um pouco mais de BPF, a gente pode manipular o comportamento que a gente quer que o sistema tenha em função da execução de uma syscall. Então nesse exemplo eu botei ali para ele encerrar a chamada e consequentemente encerrar o programa. Vocês podem ver que [interrupção no áudio] onde está a syscall e na linha 7 eu botei o que eu quero que aconteça, né? Então eu quero que a chamada encerre, que vai acabar gerando o encerramento do

programa. Posso fazer a mesma coisa com uname(F) que está ali. E, bom, e aquilo ali era meio difícil, mexer com essas chamadas aqui não é muito simples. Imagina que você tem uma lista de chamadas permitidas que esse código BPF vai ficar um pouco caro de se manter, você vai ter muita fricção, vai ter muito trabalho para manter ele, isso pode ser uma barreira, né?

Então vamos ver uma alternativa a isso. Se a gente usar [ininteligível] Seccomp, que é uma biblioteca open source também, a gente pode utilizar uma tradução do nome de uma syscall diretamente através de métodos que já estão construídos, onde a gente pode [ininteligível] de uma forma mais [ininteligível] a ação que pode ser tomada. Então a gente pode observar na primeira linha que eu estou definindo a ação padrão ali, que é permitir, e se vocês derem uma olhada na linha 4, é onde eu adiciono uma regra. E quando eu adiciono essa regra, eu falo na linha 6 o que quero que aconteça e na linha 7 eu posso dar um nome da syscall ali em texto. Dessa forma, fica bem mais fácil escrever política, na verdade, a sua política vai ser um conjunto de linhas 4 que nem está ali, você vai botar para cada syscall que você quer. No caso, se você for fazer uma abordagem [ininteligível], você vai botar o padrão como negar e vai botar várias repetições na linha 4 dizendo quais são as syscalls que você quer permitir.

Um dos problemas que a gente tem que resolver é que o nosso software muitas vezes ele já está rodando ou a gente não consegue ele o suficiente para aplicar essas políticas de filtragem. A gente precisa de uma forma de pegar no pior caso um binário que já está compilado, por exemplo, e mapear os syscalls que ele executa. Afinal, a gente não quer usar isso e parar o nosso... parar a funcionalidade do nosso sistema. Tem várias maneiras de mapear esses syscalls, uma dessas maneiras é usando o strace. O strace é uma ferramenta que você pega, você pode passar alguns parâmetros e vou ter alguns de exemplo. E o seu binário [ininteligível] ele vai fazer simplesmente executar ele e ver as syscalls que o seu binário faz isso, as syscalls do seu binário executa. Ali eu deixei em destaque a principal syscall que é a getuid. Em aplicações mais modernas, quando elas são baseadas em contêiner, baseadas em [ininteligível], a gente tem ferramentas que possibilitam a geração das políticas de Seccomp que nem ferramenta zaz faz. Eu coloquei ali, tem um repositório dela embaixo do slide. Essa ferramenta para binário, ela só funciona para binário na linguagem [ininteligível], mas você pode passar [ininteligível] também uma imagem [ininteligível], e ela vai gerar o perfil do que foi executado naquela imagem doc(F). Um ponto importante aqui é que esse formato de divisão é um formato que o [ininteligível] ali, que o [ininteligível] dele espera para aplicar essas políticas. Não é o mesmo da biblioteca que a gente vai lidar aqui. E também vocês podem reparar que no slide

que essa ferramenta não é tão precisa quando a gente fala de binário. Mesmo sendo só [ininteligível] vocês podem reparar que no binário que era um getuid ele não mapeou o getuid, ele mapeou várias outras syscalls, mas a getuid ele não mapeou.

Bom, a gente também pode querer comparar o resultado do strace com o resultado do [ininteligível], só para vocês verem como algumas syscalls são mapeadas por um e não são mapeadas em outro. O grande ponto aqui é que esse cara não é para resolver o problema de mapear as syscalls, ele é o início, né? É uma forma da gente começar a ter ali uma impressão de quais são as syscalls que estão sendo chamadas. Bom, para isso, a gente vai dar uma olhada. Tem um ponto, que é: poxa, tudo bem, eu já mapeei as syscalls, que no binário eu mapeei grande parte delas.

O que eu vou querer fazer agora é conseguir utilizar o Seccomp com filtros, tendo o menor impacto e também eu vou precisar incluir essa Seccomp durante a execução do meu binário. Para a gente poder incluir esse perfil de mapeamento de filtragem de syscalls, a gente tem que ver como o processo inicia depois que o [ininteligível] do Linux faz o trabalho de iniciar a estrutura de processo. O binário lá LF(F), ele é inicializado de uma forma bem semelhante ao que está mostrado aqui. Esse é apenas um diagrama ilustrativo, algumas das chamadas, elas podem estar desatualizadas. Desculpa ter esse asterisco ali do lado. O que a gente quer é injetar nossa política de Seccomp um pouco antes do código principal do programa iniciar. Isso é desejável porque a inicialização do processo, quando ele vai carregando outras bibliotecas e tudo, pode envolver algumas syscalls que só são chamadas ali naquele momento de inicialização, não são chamadas depois. Não são chamadas na lógica principal da aplicação. Então quanto mais próximo do fim ali onde está a main, é melhor para gente.

Então uma das formas de fazer isso, vocês podem reparar em destaque, é o preload, ter uma função de preload bem próximo de quando eu invoco a [ininteligível] principal. A gente pode usar ela para injetar uma biblioteca, por exemplo, que carregue as políticas e as syscalls permitidas pelo binário. E ela é bem acessível, você consegue configurar através de uma variável de ambiente como a LD_PRELOAD. Isso é bem semelhante à forma com que a Libsafe, que foi a primeira a primeira biblioteca a aparecer no nosso contexto histórico, funcionava. Para usar o LD_PRELOAD, a gente pode usar uma biblioteca que tenha um construtor, assim, uma função que seja invocada dentro desse construtor seria capaz de inicializar o setup do Seccomp e aplicar as políticas aí.

No exemplo aqui, eu estou colocando como seria esse construtor, né? Só para vocês verem, ele tem uma função especial [ininteligível], um construtor, ele vai ser executado antes da função principal. E o que

ele faz é justamente configurar o filtro. Esse código não é meu, a referência dele está ali embaixo. Então como esse código que estava ali embaixo já fornece uma biblioteca, então o que a gente precisa fazer é pegar essa biblioteca e colocar no binário nosso, que é o que eu estou fazendo ali na linha 1. Então na linha 1 eu faço um [ininteligível] adicionando uma biblioteca no meu binário. E logo depois vocês podem ver na linha 4 que essa biblioteca foi incluída. E como que essa biblioteca funciona, né? Ela funciona, configuro ela através de variável de ambiente. Se eu quiser ter uma abordagem [ininteligível] eu faço que nem eu fiz na linha 6. Na linha 6 eu botei uma variável de ambiente, passei para ela como parâmetro syscall que eu quero bloquear. E é claro que o meu binário getuid não tem syscall [ininteligível], então ele funcionou normalmente. Se vocês derem uma olhada agora na linha 10, eu fiz a mesma filtragem do tipo denial list, sendo que eu utilizei a syscall getuid. E vocês podem ver que, bom, a execução do meu binário foi interrompida.

Bom, para isso, se eu quiser mudar a minha arquitetura, aqui eu estou usando [ininteligível] list, se eu quiser mudar a minha arquitetura [ininteligível] list, eu preciso de uma forma um pouco melhor de passar essas syscalls. A biblioteca fornece para a gente, não preciso passar todas as syscalls que não são permitidas, eu posso passar só as syscalls que são permitidas, e ele vai bloquear qualquer outra chamada. Eu peguei esse binário aqui, peguei o resultado da strace, por exemplo, e rodei [ininteligível]. E vocês podem ver que tem uma lista, acho que eram 14 ou 15 syscalls, a minha biblioteca foi incluindo todas elas ali, e logo depois executou o binário normalmente.

Um ponto que é importante aqui é que não sei se vocês lembram, mas [ininteligível] está injetando a biblioteca logo antes da função principal do binário aqui do exemplo. Então, o que acontece? Quando eu executo strace eu pego as syscalls chamadas, inclusive aquelas que são usadas na inicialização e apenas na inicialização. Então, é muito provável que estou liberando mais syscall do que o meu programa precisa usando a biblioteca. Tem várias abordagens para saber quais são os syscalls que são utilizados na função principal. Você pode ir tirando uma a uma e ver qual funciona ou não, mas tem uma abordagem um pouco mais interessante. O Seccomp, ele produz, ele tem um modo que eu posso habilitar que é o modo [ininteligível]. O que é o módulo [ininteligível] vai fazer? Ele vai pegar as syscalls e vai permitir a execução dela e vai gerar um evento [ininteligível] Linux, de modo que eu consigo capturar quais foram as syscalls executadas.

Só para a gente ver um exemplo agora de como isso seria. Então eu botei uma lista que não permite nenhuma syscall, eu coloquei ali a ação dele, a ação padrão como log(F) e executei o binário, e vocês podem ver que foram chamadas quatro syscalls ali. Essas syscalls são as únicas quatro syscalls usadas ali pelo binário. E o que os syscalls

são? A 102 ali é a getuid, que é a nossa principal, a 5 é F start(F), e o 1 é [ininteligível] que eu faço um print na tela e a 231 é a syscall de saída do programa. Podem reparar que os syscalls estão como números, porque dependendo da arquitetura que você está usando, esse número pode mudar. Um ponto que é importante aqui é a [ininteligível] Seccomp, ela já faz esse trabalho para a gente de traduzir no número. Então a gente só referencia ali com o nome da syscall que a gente quer e ele vai traduzir para o número para a arquitetura correta.

Bom, a gente fez isso na linha 1, como vocês podem ver, primeira linha, botei como [ininteligível], vocês podem ver que não acontece nada. Poderia até apagar o default action(F), que no caso quando eu boto na lista de syscalls permitidas o padrão é negar(F). É importante só lembrar que o strace não estava errando, o que estava acontecendo é o strace pega a execução desde o início, e a nossa biblioteca só precisa das syscalls que estão ali na parte final, as syscalls que são chamadas depois do preloud para o binário que a gente quer.

Bom, talvez esse formato que esteja aí, esses logs de auditoria, dá para ver que ele é parciável(F), você pode ver que tem ali um conjunto chave valor. A gente pode ver que o tipo de evento de auditoria sempre é o mesmo. Esse evento 1326 são os eventos gerados pela Seccomp, então a gente pode confiar nisso. Sendo que a gente quer automatizar, no final das contas, o que a gente quer fazer é fazer isso de forma automatizada. Então, a gente pode mandar isso, por exemplo, para um sistema de [ininteligível] que vai executar uma ação toda vez que essa mensagem for recebida. Mas como eu disse também, [ininteligível] depois, né? A gente vai ter que ficar ligado no passe desse formato e conseguir mandar isso para algum lugar que seja relevante. Talvez aí esteja uma parte do desafio, que é transformar esses eventos de auditoria em eventos que possam ser consumidos por outro sistema, um sistema de detecção, um sistema de alerta, algo que vai abrir um incidente, ou pensando no mapeamento das syscalls em alguma coisa que vai te ajudar a mapear essa lista. A gente também não precisa inventar a roda aqui, o pessoal do Slack fez um sisteminha que é o go-audit, que o que ele faz? Ele recebe as chamadas [ininteligível] e gera esse formato [ininteligível]. E você pode mandar ela para vários lugares. Eu inclusive estou trabalhando e fazendo de ponte http(F), por exemplo, pode mandar elas como um request(F). Você pode mandar também para sislog, pode mandar para um servidor [ininteligível], no formato guelf(F), tem várias possibilidades aí. Então repara que o go-audit pode ajudar bastante nisso, transformando aqueles eventos em [ininteligível], fazendo com que você consiga, por exemplo, fazer uma filtragem muito mais rápida por tipo, mas ele não faz todo o trabalho de [ininteligível]. Você pode olhar ali que é o data

ali é o que tem a mensagem principal e você ainda vai ter que parcear(F) usando uma [ininteligível].

Bom, agora que a gente já conhece essas ferramentas, como a gente pode pensar numa arquitetura que possibilite a gente fazer essa filtragem de forma automática, com a menor fricção? A gente pode começar pegando nosso software, né? E aí peguei um [ininteligível] ao redor dele, porque eu apliquei essa biblioteca nesse software, essa aplicação está lá [ininteligível] com a biblioteca. Eu fiz diretamente do binário, não precisei me preocupar em mexer no código dela. Eu também posso pensar que a minha biblioteca, ao invés dela ler de variável de ambiente, que é algo que talvez seja difícil de configurar, ou talvez não tenha tanta flexibilidade, essa biblioteca pode consultar [ininteligível] http para obter a lista de syscalls permitidas. Então eu teria uma API e a minha biblioteca vai bater lá, fazer um GET/policies e vai obter a lista de syscalls permitidas.

Bom, uma vez que isso acontece, eu tendo uma API para fazer isso, eu posso imaginar, eu posso rodar isso no meu ambiente de homologação, por exemplo, para poder mapear syscall. Então o que vai acontecer é que no meu ambiente de homologação vou rodar [ininteligível] com log, [ininteligível] ambiente de homologação é seguro, que ele tenha todos os testes [ininteligível] necessários para isso. E, bom, você vai querer pegar os eventos de auditoria e processar de volta e mandar eles para API logo depois. Uma forma de fazer isso é usando o próprio go-audit. Então o go-audit vai se conectar no [ininteligível], vai receber os eventos de auditoria, vai transformar esses eventos [ininteligível] e depois o go-audit pode mandar esses eventos para API.

Tem outras estratégias aqui, mas tem alguns pontos de atenção. Essa estratégia precisa de cuidado, porque se você implementar ela no seu ambiente de homologação, por exemplo, é importante levar em conta que alguns componentes do software podem estar [ininteligível]. Ou seja, se o [ininteligível] não executa as mesmas syscalls que o ambiente de produção, você vai estar deixando de ver algumas coisas. Você também precisa ter uma boa cobertura de testes [ininteligível] para seguir essa estratégia. Outro ponto que você pode tentar fazer é fazer uma estratégia de kernel(F), ou seja, pode subir uma instância só da sua aplicação com log e passar uma porcentagem de usuários muito pequena para ela. Mas lembra que isso é perigoso, normalmente testar com funcionários, com o público interno seja melhor nesse ponto. E é claro, depois do go-audit, você vai querer pegar esses eventos que foram detectados e mandar para API, de modo que a aplicação, quando ela reiniciar, ela pegue uma política de syscalls atualizada.

E, bom, assim, a gente tem uma arquitetura onde a gente pode utilizar essa aplicação e a gente tem um ciclo que pode ser realimentado no ambiente confiável ou no seu ambiente de testes, no seu CI e no ambiente de produção que pode ter uma política mais restrita. E assim a gente vai ter a capacidade de detecção. Então se acontecer algum problema, alguma syscall não for admitida, o evento [ininteligível] vai ser gerado, então você vai conseguir detectar. Olha, no meu ambiente de produção essa syscall foi chamada, essa aplicação não chama essa syscalls. Você vai conseguir detectar isso na hora. Não exatamente na hora, porque o go-audit tem um delay de um segundo, vai conseguir detectar isso. E você também protegeu, está se defendendo, a sua aplicação tem uma lista ali das syscalls que ela precisa.

Tem algumas possibilidades futuras, né? Recentemente tem agora uma act de notify, não exatamente uma act, mas você pode passar um [ininteligível] e aí toda vez que uma syscall não permitida for chamada, por exemplo, você pode pedir para sua líder, pedir para a líder tomar uma ação. Então eu poderia eliminar o go-audit do meio do campo. Uma outra possibilidade é se vocês me procurarem, tiverem interesse, a gente pode testar mais e abrir o código depois. E finalmente esse ano o pessoal do Google implementou KRSI, que é uma instrumentação de Kernel que você consegue ver o comportamento um pouco melhor e usa EBPF(F). Teve uma pergunta uma sobre o EBPF(F). Então o EBPF(F) é mais flexível ainda do que o Seccomp com EBPF(F).

Tem algumas referências que são interessantes, a gente já está em cima do tempo, mas elas ficam aqui para registro. E, bom, se alguém quiser, achou legal, achou que faz sentido ter essa camada extra no seu ambiente, eu botei uma lista de ferramentas de open source que podem ajudar vocês, dependendo se o ambiente usa [ininteligível], se não usa [ininteligível], se você quer escrever uma política, se não quer [ininteligível] uma política direto. E, bom, é isso, estou aqui para dúvidas, e se alguém tiver vergonha de perguntar durante o evento, me chama no chat no LinkedIn, está de boa.

SR. ADRIANO CANSIAN: Legal, Manoel, muito obrigado. Bacana, muito legal a apresentação. Acho que temos um tempinho aí rápido para algumas perguntas. Rubens.

SR. RUBENS KUHL: Primeira pergunta é do Magno Logan, que tinha visto sobre a BPF(F), mas o detalhe que ele chama a atenção é de containers. Que ele perguntou se você já chegou a usar BPF(F) para monitorar e filtrar syscalls nos processos de containers [ininteligível] também, e se sim, como isso é feito.

SR. MANOEL DOMINGUES JÚNIOR: Eu não cheguei. Eu cheguei a usar pouquinho de BPF(F), mas para outras coisas, coisas

relacionadas a rede. Isso tem muitos exemplos na Internet de como usar BPF(F) para ver conexões de rede, para poder ver tempo de conexão, esse tipo de coisa, mas o próprio Seccomp não se importa BPF(F). Pelo menos não suportava até pouco tempo. Então o uso de BPF(F) com o Seccomp é um pouco restrito. Tem como usar BPF(F) com outras coisas. Essa minha última referência que eu trouxe, que é o que o pessoal do Google fez, KRSI, ele já suporta BPF(F). Já suporta várias outras coisas também, para você ter uma detecção mais apurada.

SR. RUBENS KUHL: Próxima pergunta do Danton Nunes. Perguntou se o processo que pode realizar uma função permitida, mas com padrão não usual, por exemplo, um servidor [ininteligível] ransomware rodando no cliente Linux, como tratar esses casos?

SR. MANOEL DOMINGUES JÚNIOR: É, o Seccomp, a filtragem dele é por syscall, né? Então pode acontecer da sua aplicação precisar de uma syscall, e se ela vier a ser explorada, ela for explorada justamente naquela syscall que ela também precisa. Isso é mais complexo. Mas é aquilo, a filtragem de syscall, ela não é uma bala de prata, ela é apenas mais uma camada extra. Tem um ponto que é importante, que é a gente continua tendo capacidade de detecção, então, malware [ininteligível] tenta usar algumas syscalls, e se elas falham, ele tenta usar outra. Ele tenta outra, ele tenta [ininteligível], depois tenta [ininteligível], depois tenta [ininteligível], ele vai tentando várias syscalls até ele conseguir o que ele quer. O ponto aí é que quando ele chamar a primeira syscall que não é permitida, o processo vai ser encerrado, e você vai ter um evento de auditoria. Ele pode fazer o [ininteligível] da memória de execução, se tiver permitido no sistema. Mas, se essa syscall estiver permitida, é mais complicado, você vai ter que se aproveitar de outras camadas de proteção.

SR. RUBENS KUHL: Próxima pergunta é do Marco Antônio Marques, ele pergunta sobre distribuições Linux utilizadas para segurança de servidores de aplicação no mercado atual.

SR. MANOEL DOMINGUES JÚNIOR: Depende muito do ambiente que você está rodando, hoje em dia se você, por exemplo, roda uma vistoria de contêiner, você pode usar algumas distribuições que têm uma superfície bem reduzida. O [ininteligível] Linux é muito famoso por isso. É bem pequeno, [ininteligível] pequenininha, [ininteligível] fortalecido com [ininteligível] security, então você consegue, você tem um contêiner pequeno que tem pouca coisa, consequentemente, uma superfície de exploração menor. Mas, assim, tem uma variedade, tudo depende muito do quanto as pessoas da sua equipe vão estar habituadas a usar uma distribuição diferente, do quanto o seu software tem de bibliotecas compartilhadas de uma distribuição ou de outra. Às vezes até mesmo a versão do Kernel(F)

porque você pode necessitar de uma versão mais antiga, uma versão que não tenha as funcionalidades que você precisa. Isso depende muito.

O que hoje as pessoas tentam fazer, para quem tem uma estrutura baseada em contêiner, é usar uma imagem bem restritiva, com bem pouca coisa para você diminuir essa superfície. E para quem usa contêiner, pode usar o Seccomp direto no contêiner. É claro que você vai, ao invés de usar uma biblioteca que vai pegar as syscalls [ininteligível] durante a execução da função principal, você também vai ter que mapear as syscalls de inicialização. Você pode ver lá no exemplo que eu trouxe do zaz, ele também tem uma lista de syscalls grande, por volta de 15 syscalls, mas isso é porque ele está pegando a syscall de inicialização que o contêiner precisa, afinal, o contêiner vai iniciar a sua aplicação.

SR. RUBENS KUHL: Obrigado, Manoel, obrigado por responder perguntas. Eu devolvo para o Adriano.

SR. ADRIANO CANSIAN: Obrigado, novamente, Manoel, sempre importante sua participação no GTS. Podemos aí, esperamos contar com você em outras oportunidades, muito obrigado.

Bom, dando continuidade ao GTS 35, versão on-line, nós gostaríamos de convidar agora a Profa. Dra. Michele Nogueira da SBC, Sociedade Brasileira de Computação, que vai falar para a gente sobre o panorama da comunidade acadêmica de segurança da informação e de sistemas no Brasil. Bem-vinda, professora, a palavra é sua.

SRA. MICHELE NOGUEIRA: Obrigada, Adriano. Só... eu compartilho minha tela ou slides?

SR. ADRIANO CANSIAN: Pode compartilhar.

SRA. MICHELE NOGUEIRA: Tá.

SR. ADRIANO CANSIAN: Perfeito.

SRA. MICHELE NOGUEIRA: Ótimo. Boa tarde a todos, é um grande prazer estar aqui. Eu agradeço ao Adriano pelo convite, espero passar uma visão um pouco mais geral em termos de cibersegurança, não apenas no Brasil, mas no finalzinho vou concentrar um pouco mais na comunidade acadêmica de cibersegurança no Brasil. Sou professora da Universidade Federal do Paraná e hoje estou representando a Comissão Especial de Segurança da Informação e Sistemas Computacionais, a CESeg, da Sociedade Brasileira de Computação.

Bom, queria começar com um certo contexto histórico, estava vendo todo o meu fundamento acadêmico, ele me leva a fazer essa reflexão que eu acho que é importante para a gente entender o nosso posicionamento em termos de cibersegurança. Se a gente analisa o mundo que nós estamos vivendo, e até mesmo passado que nós

tínhamos como grandes revoluções dentro do nosso caminho até chegar à digitalização que nós temos hoje. No início era um mundo totalmente mecanizado. Nós tínhamos propulsão por água, propulsão a vapor, essa era a força que a gente tinha, medidas baseadas na força do cavalo, que até ilustrei para vocês, e toda essa evolução foi acontecendo, até o momento que nós começamos a fazer uma produção em massa. Agora o mundo com a produção em massa, essa produção alinhada com todo o fundamento de Ford, linha de montagem, o crescimento de toda essa produção, e com ela veio o surgimento da eletricidade. E essa eletricidade, ela foi um outro ponto ali revolucionário dentro dos nossos tempos, e impulsionando, dando a base ao mundo que nós conhecemos um pouco mais de perto hoje, que é o mundo dos computadores e da automação das nossas atividades. Então iniciando ali com o próprio mainframe, e a evolução desses computadores para os computadores pessoais. E hoje computadores vestíveis, computadores que estão em qualquer lugar e a qualquer momento do nosso dia a dia. E isso impulsionou também a automação. Então nós passamos ali de uma linha de montagem com a força de trabalho humano, capital humano, para automatizar essas atividades em termos de indústria.

Bom, por que eu estou falando tudo isso? Porque a gente chega hoje no momento em que nós encontramos em um mundo totalmente conectado, onde essa evolução aconteceu e cada vez mais o que a gente busca dessa conexão e da inteligência. Alguns sistemas são considerados até inteligentes, apenas por estarem conectados à Internet, onde ele pode ter informações, pode obter dados e também oferecer dados para essa conexão, para esse sistema todo conectado. Também, obviamente, a gente precisa falar aqui, quando fala de inteligência, do uso das próprias técnicas de inteligência artificial para associar e conectar esses dados de fato de uma forma que faça sentido e que possa apoiar esses sistemas como um todo. Aqui só um outro exemplo, antes de passar para uma ideia geral dessa revolução, seriam os próprios carros autônomos e toda a centralização através dessas técnicas de IA que nós temos, só como ilustração aqui também da evolução desses sistemas. Então nós começamos lá com aqueles sistemas gerados a partir das forças da água, força do vapor, passando pela eletricidade, a evolução desses sistemas, para chegar nesses sistemas inteligentes e conectados que nós temos hoje, que vêm, inclusive, revolucionando as nossas indústrias, chamando da própria indústria 4.0, a geração de uma grande quantidade de dados e também uma certa flexibilidade nos nossos sistemas.

Tudo isso, esses pontos que eu quis ressaltar aqui para vocês são reforçados dentro dessa ilustração que eu coloco agora, que são exatamente as chamadas revoluções industriais. A primeira, a mecanização, que é o que eu ilustrei inicialmente com a força da água,

força do vapor, etc. A segunda com a produção em massa e a linha de montagem através do surgimento da eletricidade. A terceira, o surgimento dos computadores, a evolução deles e a automação que eles permitiram. E hoje que é a chamada 4ª Revolução Industrial, que é que nós nos encontramos, essa transformação, ela vem sendo feita, que é através dos sistemas chamados de ciberfísicos ou ciberhumanos, com a integração desses dispositivos computacionais dentro dos nossos ambientes e dentro, às vezes, até mesmo do nosso corpo, não exatamente dentro ou o uso dos dispositivos no nosso corpo, tendo essa integração com todos esses sistemas. Então eu trouxe essa sentença de Klaus Schwab, que foi exatamente quem cunhou esse termo de Quarta Revolução Industrial. Ele era, na época, o fundador do Fórum Econômico Mundial, ele é o fundador do Fórum Econômico Mundial, na época era presidente do Fórum. E ele coloca, né, que essa ubiquidade, supercomputadores, a mobilidade, os robôs inteligentes, os carros autônomos, essas melhorias. A revolução que a gente pode fazer na própria genética, editando a genética através dessa revolução tecnológica que vem acontecendo. Essas evidências, né, elas são dramáticas, elas estão no nosso entorno e a tendência é que isso, essas mudanças aconteçam em uma velocidade exponencial, a partir de agora. E isso é o que a gente vem vendo, ao longo nos nossos dias, dentro da nossa sociedade. E essa revolução, poderia até acrescentar aqui dentro da mensagem dele, a própria pandemia que acelerou ainda mais esse processo de digitalização de uso desses próprios sistemas.

Então, vendo o nosso presente e olhando um pouco mais para o futuro. O que a gente vê em relação a tendências da internet, claro que a gente tem toda uma infraestrutura que dá suporte a essas aplicações mais de borda, mas o que a gente percebe são algumas características que se espera que elas sejam impulsionadas nos próximos anos. Como a questão da ubiquidade, ou seja, você ter o uso desses dispositivos de forma cada vez mais transparente para você. É um pouco o que acontece já com os nossos próprios telefones portáteis ali, os seus smartphones, você, hoje, você acaba não conseguindo ficar sem eles para realizar suas atividades. A diversidade desses dispositivos, então, a gente tem uma imensidão de dispositivos, todos ou grande parte deles, conectados à internet. Então, desde celulares, mas televisões, refrigeradores, computadores pessoais, câmeras, relógios, tablets, tudo isso conectado, tudo isso coletando informações nossas e gerando informações nossas, para esses ambientes. Que é exatamente o que é colocado como a geração de uma grande quantidade de dados, que seria o *big data*.

Então, quando a gente analisa um pouco mais longe, né, existe até uma tendência que se alguns futurologistas em termos de tecnologia, eles colocam que nós chegaremos em um ponto que é considerado o ponto de singularidade, onde a capacidade

computacional que nós teremos, ela vai superar a capacidade humana em termos de inteligência. Se isso realmente vai acontecer, é a grande discussão que se tem hoje. Mas o que eu quero passar aqui, em termos de integração desses dispositivos no nosso dia a dia, né, e aqui, como está sendo ilustrado, até mesmo no nosso corpo. De forma que, em um certo momento, a gente não vai saber mais a distinção entre o que é, de fato, dispositivo computacional e o que é, de fato, o humano, não é? Como é que a gente consegue separar esses dois elementos.

Eu tenho um vídeo, eu não sei se vai sair o som. Eu vou tentar, senão eu vou saltá-lo, mas é apenas ilustrativo de todo esse contexto que nós estamos passando. Vocês vão entender daqui alguns minutos porque eu estou falando de tudo isso em relação à cibersegurança, mas eu queria tentar ilustrar aqui, um pouco mais a questão da... Deixa eu ver se eu consigo.

Então, esse vídeo, ele vai dar uma perspectiva desse nosso dia a dia, e um pouco mais dessa previsão ou predição em relação à evolução dos nossos sistemas.

[exibição de vídeo]

SRA. MICHELE NOGUEIRA: Bom, o vídeo, apesar de estar sem áudio, eu acredito que, como ele é bem ilustrativo, os textos todos. O áudio era só para ler os textos que estavam sendo apresentados.

Bom, a ideia de apresentar esse vídeo é exatamente do que espera, né, dessa transformação que nós estamos passando. E essa transformação toda, o que ela tem a ver com relação à cibersegurança? No fundo, ela tem tudo a ver com a cibersegurança, porque, quando você imagina, naquela primeira etapa, naquela Primeira Revolução Industrial que eu mostrei para vocês, qual era o interesse que nós tínhamos de nos proteger? A gente nem tinha sistemas computacionais, a ligação, a conexão entre esses sistemas, na verdade, não existia, até um pouco no início daquela geração da Terceira Revolução Industrial, com o surgimento dos computadores, onde os computadores ficavam todos guardados em uma sala sem ter conexão com outros. Esses sistemas, eles, talvez, as principais questões de segurança eram mais relacionadas à segurança física daqueles dispositivos, e não necessariamente à segurança no sentido do ciberespaço, até porque esse ciberespaço não existia. E uma vez que a gente vai integrar todos esses dispositivos, a nossa preocupação, então, com cibersegurança, ela é proeminente. Porque agora nós precisamos, realmente, nos preocupar com esses sistemas. Nós precisamos nos preocupar com esses dados que são, estão sendo trafegados nos nossos sistemas.

Aqui eu coloquei uma ilustração em relação não apenas à segurança, mas à privacidade também, desses dados. Desde as redes, né, as redes de acesso que seriam mais próximas ali e a interação com os usuários, enfim. Tanto o acesso a esses sistemas, a essa rede como a informação, a proteção à informação em si, ao dado em si, ela precisa ser preservada. E agora a gente, uma vez que eles estão conectados, esses dispositivos aqui das redes de acesso, elas estão conectadas, elas têm vulnerabilidades. E a gente sabe, muitas vezes, que esses dispositivos e os softwares que são projetados para eles e implementados para esses tipos de dispositivos, muitas vezes, nós sabemos que as companhias, elas não têm uma preocupação prioritária com segurança. Isso não acontece só com as redes de acesso, mas elas acontecem em todo esse caminho, desde a rede acesso à rede, o ponto de conexão, aqui, o *gateway*, a própria *cloud* e os data centers nas instituições. Então, todo esse caminho aqui, todo esse ciclo onde nós temos transmissão de dados e a conexão entre esses elementos, eles requerem, então, o nível de segurança, e requerem também, hoje com as próprias regulações que nós temos em relação à privacidade, elas vão requerer cada vez mais, uma atenção maior à questão de privacidade dez dados, tá?

Então, o que muda hoje? Com a Quarta Revolução Industrial, é esse tipo de sistema que nós temos, é essa interconexão entre esses dispositivos e esses sistemas que estão ali sendo executados. E tem um outro ponto que é relevante aqui, que é a questão da heterogeneidade dos dispositivos. Se nós também compararmos com a Terceira Revolução Industrial, que era aquela que começou com *mainframe*, os computadores pessoais, nós poderíamos, nós tínhamos uma certa uniformidade de que tipos de dispositivos nós teríamos conectados. E hoje, já essa heterogeneidade de dispositivos é muito grande. Como eu falei, ela vai desde o nosso relógio que está conectado, até uma geladeira, uma televisão e a um próprio data center que nós temos. Então, a complexidade em projetar a segurança para esses dispositivos é cada vez maior. Então, a gente precisa começar a olhar, né, para esses dispositivos e começar a gerar essas soluções de segurança de uma perspectiva diferente do que a gente tinha até então. Porque a revolução, essa nova etapa que nós estamos passando, com todas essas ilustrações que eu mostrei para vocês, ela é diferente, então, ela traz um sistema que requer, ela tem requisitos diferentes em relação à segurança.

Aqui um outro exemplo, mais voltado a esses dispositivos ali no centro, a questão dos sistemas, as redes veiculares e os tipos de ataques que nós temos em relação a esses dispositivos. Então, ali você pode ter modelos de ataques que vão tentar controlar o carro, ou abrir a porta do carro, ou destravar a porta de um carro remotamente, seria um modelo de ataque aqui dentro desse contexto. Poderia ter um outro

tipo de ataque que vai tentar roubar informações desse veículo e dos passageiros que estão ali nele, ou até eventualmente também, comprometer os serviços, toda a engrenagem desse carro que está em movimento. E o que acontece é que você tem situações que pode, inclusive, colocar a vida da pessoa em perigo, não é? A vida dos passageiros desse automóvel em perigo, com um tipo de ataque desse.

Então, vejam que aí, por si só, você já tem uma mudança no nível de comprometimento que os atacantes, eles podem agora chegar, o que não acontecia até então. Aqui no lado esquerdo, vocês estão vendo a ilustração de uma casa, todos os dispositivos eletrônicos que possivelmente estão conectados, ali, uma lâmpada inteligente ou até mesmo um alarme, a geladeira. Então, também temos, podemos vislumbrar modelos de ataque para esse tipo de cenário, não é? Um ataque que vai controlar a luz da sua residência, que pode, inclusive, utilizar esse tipo, essa entrada ou alguma vulnerabilidade que possa ter dentro desse dispositivo para comprometer a casa inteira. A mesma coisa acontecendo ali com o alarme. O roubo de informação em relação ao alarme e isso pode ser usado, por exemplo, por assaltantes da sua casa, que vai saber quando que você está em casa ou não, e o momento certo de atuar nela. Então, isso tudo é através da internet. E a geladeira ali também, para comprometer ou enviar *spams* para você fazer compras de produtos associados ao teu dia a dia. O que por si só ali, já pode comprometer a própria privacidade do comportamento da tua casa.

Aqui do lado direito, já ilustra dispositivos vestíveis. Uma rede ali, uma rede vestível, onde você tem uma pessoa com vários dispositivos, utilizando vários dispositivos, desde um dispositivo que vai fazer um acompanhamento das suas atividades físicas. O hacker também pode utilizar essas informações, roubar essas informações para entender o teu comportamento. Pode comprometer, eventualmente, um *pacemaker* que você esteja usando. Você tem problemas cardíacos, ele vai, remotamente, e compromete isso. Que poderia também, colocar em risco a vida da pessoa. E outros tipos de situações de controlar, por exemplo, que já aconteceu esse tipo de ataque, de controlar uma bomba de insulina que o usuário estava usando. E o atacante conseguiu, remotamente, através de algumas vulnerabilidades de software, controlar aquela bomba de insulina e, eventualmente, aumentar mais o nível de insulina que estava sendo liberado, ou reduzia a quantidade que seria necessária para aquele paciente.

Então, em todas essas situações, quando você coloca na perspectiva do usuário e dessas redes vestíveis, a situação fica ainda mais crítica, porque acaba comprometendo diretamente, ou pode comprometer diretamente a vida das pessoas. Então, mais uma vez, vejam que agora, os novos modelos de ataque que nós temos, eles

começam a ter um nível de comprometimento, né, um nível, realmente, de colocar o usuário em risco, muito maior do que o que a gente tinha antes.

Aqui, outras ilustrações desses tipos de ataques, colocando e, uma perspectiva da nossa pandemia e do trabalho de casa. Nós acabamos esquecendo de proteger algumas partes do nosso sistema, então, o fato de nós estarmos em casa, trabalhando de casa, começa a chamar a atenção de alguns atacantes, para comprometer a privacidade dos dados, que nós estamos ali, trafegando, na nossa rede doméstica. Eventualmente, são dados do trabalho, então, começam a ser dados mais sensíveis ali, que você está sem, muitas vezes, a proteção que você teria no escritório. E abre outras também, outros alertas em termos de vulnerabilidades de outras partes da tua casa, que podem ser vulneráveis. E lembrando, inicialmente, esses dispositivos acabam não... que nós estamos utilizando hoje, através dessa revolução, a segurança, ela ainda não é vista como uma prioridade para essas empresas. E a gente sabe também que existe muitas falhas, né, falhas de desenvolvimento do software, falha de desenvolvimento do hardware. E são esses pontos que muitas vezes, são vulneráveis aí, e sujeitos à ação desses hackers.

Aqui um outro exemplo, também colocando essa perspectiva da pandemia e o trabalho remoto, de casa. E de você ter profissionais ou pessoas que se passam por outras para te vender algum produto, ou até mesmo apenas para obter as suas informações. Isso também foi um ponto associado com a privacidade dos dados e os riscos que aumentaram com esse trabalho remoto.

Então, aqui, há um outro exemplo de vírus que, uma outra vulnerabilidade ali, para encontrar algumas brechas de segurança em relação também, a Covid-19. Claro que aqui tem uma ilustração do vírus, mas a representação aqui, ou a associação que se deseja fazer aqui com essa ilustração é a criação, é a geração de vírus para sistemas computacionais associados, ou que aumentaram diante da nossa situação de pandemia.

E um outro ponto que também é associado com botnets, com essas redes de dispositivos infectados, são as características novas que nós começamos a observar com essa revolução que nós estamos passando. Então, antes, quando a gente compara com o que é chamado de botnets tradicionais, esses dispositivos todos eram... as características mais comuns que eram dispositivos fixos, que você sabia ou que você conseguia identificar facilmente a localização, ou teria pelo menos uma ideia da localização, com base no IP. Entretanto, essa nova geração de botnets, inclusive, que foi usada em alguns ataques recentes, elas são, que são chamadas botnets móveis, elas se utilizam desses dispositivos, dos dispositivos vestíveis, dos dispositivos

móveis, portáteis. E logo, a complexidade de você identificar essas botnets é muito maior. Então, também alterou as características das botnets quando a gente compara com botnets tradicionais.

Então, os dispositivos, são três principais características que nós percebemos aí, nessa mudança. Uma delas é a diversidade dos dispositivos, agora a diversidade é maior. A localização desses dispositivos já não é tão simples de você identificar, até porque esses dispositivos podem, né, como eles são móveis, eles podem estar, podem ter uma flexibilidade em relação à localização. E um outro ponto é com relação à quantidade desses dispositivos que fazem parte da botnet. Então, agora a gente percebe que os atacantes, eles conseguem recrutar uma quantidade muito maior desses dispositivos para participarem da botnet. Então, também trazem desafios na detecção dessas botnets e até mesmo na contenção dos ataques que elas podem prover, como, por exemplo, os ataques de negação de serviço.

Então, dentro desse contexto inteiro e dessas transformações que vêm acontecendo, eu vou falar agora para vocês, um pouco do trabalho que nós fazemos dentro da Comissão Especial de Cibersegurança, da Sociedade Brasileira de Computação. De uma forma bem resumida, a comunidade, ela trabalha em uma perspectiva de seis eixos, que são seis grandes desafios da cibersegurança no Brasil. Em 2014, pesquisadores da área, eles se juntaram, e a pedido da SBC, foi criado um documento em que se levantaram os grandes desafios da cibersegurança no Brasil. E o que se chegou foi a esses seis eixos que vocês estão vendo aqui: a gestão de segurança da informação; resiliência e tolerância a sistemas críticos; teoria da informação, criptografia e algoritmos; gestão de identidades; segurança de redes, serviços, sistemas ciberfísicos e ciber-humanos; e a forense digital. Então, quando a gente olha para a nossa comunidade de cibersegurança, vou falar um pouco mais à frente, um pouco mais de detalhes dessa comunidade. Mas quando a gente olha em termos acadêmicos, as atividades que vêm sendo realizadas, elas se encaixam nesses seis eixos, ou nesses seis grandes desafios da cibersegurança no Brasil.

Então, a comissão especial, ela foi formada por membros da Sociedade Brasileira de Computação, foi criada em 2004. Quem tiver interesse de participar, nós temos uma lista de discussão, a seg-l. Tem o link aqui para página da CESeg. Eu vou repetir esse link no final, para quem tiver interesse em guardar. Também vou apresentar aqui para vocês, rapidamente. E, por favor, fiquem convidados, se sintam convidados para participar da lista. O nosso viés é um viés mais acadêmico, então, são professores, pesquisadores da área que participam dessa lista de discussão. Mas existem algumas ações que eu vou falar também em alguns slides, que talvez seja interessante

para fazer a ligação entre essas duas comunidades, não é? A CESeg e essa que vocês participam.

Então, das principais atividades da nossa comissão, estão, hoje, a organização do Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais, o SBSeg. Muitos de vocês podem, talvez já tenham participado. O fomento, a interação e a integração dos grupos de pesquisa em cibersegurança no país. E o Instituto Nacional de Segurança Cibernética, o INCT-Seg.

Então, eu vou falar um pouquinho de cada uma dessas atividades. O Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais é um fórum essencialmente de pesquisadores e integração com a indústria, estudantes e profissionais da área de segurança. Ele atua dentro daqueles, dos seis eixos que eu mostrei para vocês anteriormente. O próximo evento vai ser em Belém, provavelmente em agosto de 2021. A data ainda pode ser modificada diante da situação da pandemia. E nós já estamos na 21ª edição da organização desse evento. Então, ele é organizado anualmente. Exceto este ano que aconteceu remotamente por conta da pandemia, mas a intenção também é ter uma aproximação física entre os participantes, não é? Essa conexão, você pode conversar, conhecer outras pessoas, interagir e eventualmente fazer outras ações no futuro.

Com relação aos grupos de pesquisa, eu trouxe para vocês aqui, uma visão que foi gerada. Esse mapa é um mapa que reflete ali os pontos onde nós temos grupos de pesquisa em segurança cibernética no Brasil. Essa imagem, ela está sendo atualizada no momento, então, eu trouxe essa ilustração mais para vocês terem uma noção do que nós temos, a visão que nós temos até hoje, mas como eu disse, ela está sendo atualizada. Então, nós temos grupos que se concentram muito mais ali no Sudeste e no Sul, não é? E grupos que eu falo, grupos de pesquisa em cibersegurança. E um grupo que está no Amazonas.

Então, uma outra atividade com relação ao Instituto Nacional de Ciência e Tecnologia em Segurança Cibernética, é uma rede formada por várias instituições no Brasil. Aqui está listado o nome de cada uma das instituições que participam. E é uma rede capitaneada pela Universidade de Campinas, em São Paulo. Nós temos algumas ações, além das ações de pesquisa, nós temos algumas ações de formação de recursos humanos, integração entre as pessoas etc., que eu vou também, mencionar um pouco mais à frente para vocês. Então, aqui só para ilustrar o selo de concessão, aqui, do instituto. É um selo que... na verdade, esse, o instituto, ele é fomentado pelo CNPq.

Então, essas são as atividades principais CESeg, da nossa comissão especial. E tem algumas ações, algumas novas atividades que, dentro da minha gestão, eu pretendo impulsionar um pouco mais forte. Que são: essa colaboração com outros grupos, eu acho que isso

é um ponto mais forte dentro aqui das atividades que eu espero fomentar nessa minha gestão. A integração com outros grupos, apoio. A criação, a geração de novos tutoriais, palestras, cursos relacionados, para a gente poder, realmente, estender o conhecimento, ampliar um pouco mais o alcance em termos de conhecimento, para a comunidade de uma forma geral.

Também listei aqui algumas outras ações, só para finalizar, não é? Caso alguém não saiba, mas todos, de certa forma, estão relacionados com a CESeg e com essas ações novas de integração. Uma que eu queria ressaltar aqui é o projeto Mentored. É um projeto financiado dentro de um edital do MCTIC/CGI/Fapesp. O que eu queria ressaltar desse projeto é a proposta de criação de um ambiente acadêmico para a experimentação em cibersegurança no Brasil. Nós percebemos que é um ambiente que falta e é necessário para que a gente possa fazer as experimentações. Existem vários projetos de pesquisa, desenvolvimento e inovação dos grupos acadêmicos que eu mostrei anteriormente. Um ponto de ação que eu quero fortalecer é a minha integração com a IEEE ComSoc, particularmente, a comissão técnica da internet. Tem algumas outras ações aqui, como a coluna de atualidades em cibersegurança na revista da SBC Horizontes, para quem tiver interesse. Também, quem quiser escrever para a coluna, sintam-se convidados.

E algumas ações que nós temos feito, que seriam, que é a série Conversando com Líderes de Cibersegurança, junto com a Escola de Redes da RNP. E uma palestra que nós realizamos recentemente, com o presidente da Sociedade de Comunicação da IEEE ECC(F), que falou essencialmente de alguns desafios de segurança. Aqui, eu ilustrei esses últimos itens, mas como uma forma de procurarmos ter essa interação com a comunidade de vocês, pensar em ações conjuntas que nos permitam trazer abrangência desse conhecimento e ampliar, um pouco mais, o escopo de integração das pessoas da comunidade, que trabalhem na área de segurança.

Então, aqui um agradecimento a vocês. Essa é a página da CESeg. E eu fico à disposição para perguntas.

SR. ADRIANO CANSIAN: Obrigado, professora Michele. Muito legal aí, as ações da comissão.

Vamos passar, então, para o Rubens, que vai fazer algumas perguntas que passaram pelo nosso chat no YouTube. Rubens, por favor.

SR. RUBENS KUHL: Obrigado, professora Michele. A primeira pergunta é do Marco Antonio Marques. Ele comenta que o Brasil tem poucos cursos focados em segurança da informação nas universidades, e que as grades dos cursos têm poucas disciplinas de segurança. E a pergunta dele é se faltam políticas públicas.

SRA. MICHELE NOGUEIRA: É. O que a CESeg tem feito junto com a SBC é a tentativa de impulsionar o próprio currículo de cursos específicos de cibersegurança. Então, bacharelado em cibersegurança, não é? Ações nesse sentido. Infelizmente, essas ações, elas ainda não avançaram muito, até porque existem pessoas que acreditam que não existe essa necessidade de ter cursos específicos de cibersegurança. É um entendimento diferente do que a CESeg tem. A gente acredita que, sim, é necessário. É um movimento que a gente vê fora do Brasil. A própria ACM, ela traz um currículo específico de cibersegurança, hoje já. Ela já oferece e recomenda, né, como é que seria essa distribuição, esse currículo. E aqui no Brasil ainda existe... e existe, o que talvez seja normal, né, você ter essas diversidades ali, em termos de visões sobre a importância da cibersegurança.

Então, a gente precisa reforçar um pouco mais isso, precisamos de ações que realmente tentem, e continuem tentando mostrar que é importante a gente ter cursos um pouco mais voltados, realmente, para a questão de cibersegurança. Isso falando de cursos no nível de bacharelado, não é? Claro que a gente pode fazer ações no nível um pouco mais técnico. Mas a nossa perspectiva seria mais ações, de fato, no curso, no nível de graduação, no nível de pós-graduação. Então, mestrado e doutorado, que é o que... que são movimentos que a gente já consegue identificar fora do país.

SR. RUBENS KUHL: O Alberto Junior fez uma pergunta, que já acabou sendo respondida durante a apresentação, mas ele fez também um comentário aí, que pode ser endereçado, de que, às vezes, a segurança mecânica ainda é mais segura em certos aspectos do que a segurança digital.

SRA. MICHELE NOGUEIRA: Sim.

SR. RUBENS KUHL: E a última pergunta, do Jair Avillez, e ele diz assim: IPv6, 5G, IoT, inteligência artificial, *cloud*, drones, carros autômatos, etc., como os usuários podem se sentir seguros quanto ao nível de segurança atual?

SRA. MICHELE NOGUEIRA: Veja, a gente precisa trabalhar bastante para mudar a visão que nós tínhamos até então. Essa visão que eu acabei de mencionar sobre os cursos, ela reflete bem, talvez, também, o que a gente perceba nas indústrias, na nossa sociedade. Nós, como usuários, também temos um papel nisso. Então, nós precisamos impulsionar essa, esse reforço da necessidade de segurança, para que a gente... As indústrias e as empresas, de uma forma geral, elas não vão olhar com prioridade para a segurança, se nós também não impulsionarmos isso. O exemplo é a própria Lei Geral de Proteção dos Dados.

Então, essa, a necessidade, eu vejo, são dos dois lados, no sentido de tentar impulsionar essa necessidade, ou fortalecer a

motivação de trabalhar, de olhar, de dar prioridade para a segurança. Como é que a gente pode se sentir seguro hoje, né? Eu, pessoalmente, não tenho essa segurança, também, tenho toda a proteção, mas a gente precisa impulsionar essas ações, para que a gente consiga transformar esses sistemas a serem mais seguros. É isso que eu diria.

SR. RUBENS KUHL: Obrigado, professora. Obrigado a quem nos mandou perguntas. Eu devolvo para o Adriano, para o encerramento.

SRA. MICHELE NOGUEIRA: Muito obrigada a vocês.

SR. ADRIANO CANSIAN: Obrigado, professora. Gostaria de agradecer a todos que estiveram conosco nessa manhã, principalmente as interações que tivemos no YouTube, não é?

E eu gostaria também de convidar o Frederico Neves para falar sobre o encerramento do evento GTER, GTS, sem esquecer que esses dois eventos, eles fazem parte da Semana de Infraestrutura da Internet no Brasil, que continua ao longo de toda essa semana. Então, se o Fred quiser...

SR. FREDERICO NEVES: Bem lembrado, Adriano. Então, estamos aqui encerrando o GTER 49 e o GTS 35. Nessa edição, a edição, a pandemia, como todos sabem, a gente não teve o evento no começo do ano, em função do começo da pandemia. A gente já está praticamente em voo, e a perspectiva é de a gente ter o evento no início do ano que vem, também é que ele seja remoto. Mas a gente espera conseguir fazer não um evento de apresentações convidadas, como esses dois, mas que a gente tenha uma chamada de trabalhos, mas, novamente, ainda em uma versão totalmente remota, para o início de 2021. É óbvio que a gente não sabe o que vai acontecer amanhã, mas a perspectiva é essa, de que a gente continue na forma que estamos efetuando o evento hoje.

Como o Adriano disse, a gente precisa agradecer a toda a equipe que ajuda a organizar o evento, agradecer os dois comitê de programa que trabalharam nos convites e conseguiram montar esses dois eventos que tiveram excelente qualidade. E agradecer os apresentadores, óbvio. E contar, novamente, com a comunidade para apresentar trabalhos agora no início de 2021.

E como o Adriano disse, a Semana de Infraestrutura da Internet no Brasil, essa é a décima semana, o décimo ano consecutivo. Ainda continua amanhã, quinta e sexta, com os eventos do IX Fórum. Adriano, você quer...

SR. ADRIANO CANSIAN: Acho que é isso. Agradecer as pessoas que cuidaram da transmissão do stream, a equipe lá do NIC, que cuidaram muito bem de toda a organização aí, esteve tudo perfeito. E esperar que no próximo ano, nessa época de dezembro, onde nós fazemos tradicionalmente o evento em São Paulo, todos nós

possamos estar bem e com saúde, e voltarmos lá para o nosso evento presencial, porque o ser humano é social, não é? Então, seria muito bom se a gente puder fazer isso de novo em dezembro de 2021. Espero ver todos vocês lá, se Deus quiser.

SR. FREDERICO NEVES: Isso mesmo. Joia. Tchau, tchau, pessoal.

SR. ADRIANO CANSIAN: Obrigado, pessoal. Até breve.

SR. FREDERICO NEVES: Até.