Prevenção de ataques *zero-day* por meio de vulnerabilidades divulgadas

Alexandre Y. Kajihara[1]

[1] DACOM - Departamento Acadêmico de Computação do Curso de Bacharelado em Ciência da Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão – PR – Brazil

alexandre.ykz@gmail.com

Resumo. *Os crimes cibernéticos são crimes cometidos através do uso da Internet ou de outras redes de computadores. Um tipo de crime que está sendo cometido é o ataque zero-day, em que se explora vulnerabilidades que não foram divulgadas. Devido ao fato dessas falhas não serem expostas, queremos auxiliar usuários que tem aplicações, podendo elas serem Web ou não, prevenir que as suas aplicações não fiquem expostas, e consequentemente que seus dados sigilosos não sejam divulgados as pessoas não-autorizadas. Para tal façanha, iremos utilizar os detalhes das falhas que foram divulgadas e a partir das informações presentes iremos utilizar as mais relevantes, ou tentar combinar informações para conseguir informações relevantes. Com as associações feitas, testar a aplicação para verificar se existe alguma falha a partir de falhas existentes, e indicar ao fornecedor de software quais ataques daqueles que foram divulgados podem ser utilizados em sua aplicação, ou seja, que não sejam vítimas de ataques que não foram divulgados. Com essa prática de prevenção desejamos que uma parte dos crimes cibernéticos diminuam, e que aplicações atuais e novas não sofram do ataque de zero-day.*

1. Introdução

O crime cibernético é uma forma de crime que inclui crimes que são cometidos através do uso da Internet ou de outras redes de informáticas [Paolo Foti 2015]. O aumento da disseminação de crimes por computador deve se a *Return On Investement* (ROI) derivado desse tipo de atividade criminosa [Paolo Foti 2015]. Existem vulnerabilidades que podem valer mais de 100.000 dólares, dependendo da sua complexidade [Leyla Bilge 2012], um exemplo que podemos citar é as vulnerabilidades do tipo *zero-day*, que iremos explicar o que elas são logo abaixo, em que devido a esse retorno de investimento os valores variam, na qual no segundo dia ela pode valer um quarto do que valia no *zero-day* e no décimo dia o valor dela vale 1/1000 [Paolo Foti 2015]. Além do que, essas vulnerabilidades são muito desejadas por agências governamentais, cibercriminosos e empresas de software [Pratap Kumar 2016].

Um dos tipos de ataque existente é o ataque de *zero-day*, em que acredita-se que o período em que provavelmente ocorreram as primeiras vulnerabilidades de *zero-day* aconteceram nos anos 80 [Paolo Foti 2015]. Esse ataque cibernético que explora uma vulnerabilidade que não foi divulgada publicamente, e esse ataques visam um número de organizações que possuem informações confidenciais que podem ser roubadas [Leyla Bilge 2012]. Essas vulnerabilidades permanecem desconhecidas até por 2.5 anos

[Leyla Bilge 2012]. Enquanto a vulnerabilidade permanece desconhecida, o software obtido não pode ser corrigido e os produtos antivírus não conseguem detectar o ataque através da verificação baseada em assinatura [Leyla Bilge 2012].

Uma maneira de prevenção é a implementação de sensores ou defesas cibernéticas, *firewall* [Last 2016]. Outras maneiras de prevenção seria por meio de *Host Intrusion Protected Software* (HIPS), em que não se depende de assinaturas, já que há um monitoramente das atividades da máquina, usar um bom antivírus protegendo de ataques conhecidos e desconhecidos e atualizar suas aplicações sempre que houver notificação de atualização de *software* [Pratap Kumar 2016]. Usuários que utilizam o antivírus podem ter mais cuidados com a segurança de seus computadores, e podem ser menos expostos aos ataques [Leyla Bilge 2012]. Existem testes de penetração como o *Web Application Penetration Testing* (WAPT) [Pratap Kumar 2016]. Apesar de todas essas maneiras de prevenir é díficil, porque a informação só estará disponível quando o ataque tiver sido concluído [Pratap Kumar 2016]. Além disso, os fornecedores de *software* dão preferência de corrigir vulnerabilidades que foram divulgadas ou estão prestes a declarar [Leyla Bilge 2012].

Devido ao fato de que nesse tipo de ataque às falhas não são expostas, queremos prevenir à partir de vulnerabilidades que já foram divulgadas para informar aos fornecedores que possuem ou que irão lançar um possível *software*, quais falhas estão ocorrendo ou já ocorreram, para que os mesmos possam prevenir e não sejam vítimas do ataque *zero-day*. Essa prevenção ocorreria por de meio de uma análise de dados, em que iriamos combinar os dados que possuem mais relevância, e em que essas informações são obtidas por meio de sites que divulgam esses detalhes, como o *National Vulnerability Database* (NVD), do *National Institute of Standards and Technology* (NIST) e o *Common Vulnerabilities and Exposures* (CVE).

Como foi dito anteriormente os crimes cibernéticos vem aumentando devido ao retorno financeiro. Acreditamos que o fato de que a cada dia termos uma nova aplicação, faz com que os crimes continuem crescendo, e com a prevenção do ataque *zero-day* e de qualquer outro tipo de ataque pode evitar com que as informações presente neles, sejam expostas as pessoas que não deveriam ter acesso. Além do que, julgamos que precaver pode impedir que tenhasse prejuízos financeiros, já que nesses dados podem conter informações de senhas de bancos. Aliás, ter essa cautela seria uma forma dos fornecedores de *software* de se defenderem desses cibercriminosos. Isso porque, sabendo de quais vulnerabilidades os atacantes estão se aproveitando, as organizações saberiam o que precisa ser corrigido ou não na sua aplicação, e consequentemente poderiam lançar atualizações para corrigir as vulnerabilidades presentes, coibindo a ação desses cibercriminosos.

2. Referencial teórico

O ataque de *zero-day* é um ataque cibernético que explora uma vulnerabilidade que não foi divulgada publicamente, e com isso não há defesa contra esse ataque, consequentemente não o *software* não pode ser corrigido e os antivírus não detectam devido a não ter assinatura [Leyla Bilge 2012]. O mesmo autor citado anteriormente, relata que identificou-se 18 vulnerabilidades no mundo real antes da divulgação em que dessas, 11 não eram anteriormente conhecidos por terem sido empregados em ataques *zero-day*,

sugerindo que esse tipo de ataque é mais comum do que se pensava.

Os ataques de *zero-day* contra aplicações Web aumentaram [Pratap Kumar 2016]. Alguns argumentos utilizados, são que fornecedores pagam US$ 60,000 dólares de recompensa na descoberta de novas vulnerabilidades em seus produtos [Serge Egelman 2013], mais de US$ 100,000 dólares dependendo da complexidade da vulnerabilidade[Leyla Bilge 2012], falhas vendidas para a *National Security Agency* (NSA) por US$ 50,000 dólares, e com esses valores crescente justificam sendo uma motivação primária para a crescente popularidades do mercados de *zero-day* [Serge Egelman 2013]. Além do que, essas práticas de programas de recompensas de *bugs* existiam já em 1995 [Serge Egelman 2013]. Um outro motivo citado são quando os *hackers* utilizam o *software* ou por *honeypots* [Paolo Foti 2015]. Esses *honeypots* seriam uma maneira de notar as ações que podem ser analisadas, observadas e compreendidas [Spitzner 2003]. Também podem os ter que as vulnerabilidades podem ser resultados da configuração incorreta de sistemas computacionais cada vez mais complexos, descuido dos usuários do sistema, como o uso de senhas fracas ou o compartilhamento de senhas com colegas e amigos [Paolo Foti 2015].

Inúmeras soluções são citadas como: gerar previsões de descoberta de vulnerabilidade nos níveis global e categoria (navegador, sistema operacional e *player* de vídeo) [Last 2016], Teste de Penetração de Aplicações na Web (WAPT), *Host Intrusion Protected Software* (HIPS), antivirus, atualizações de aplicações, navegadores e bloquear anexos de arquivo de e-mail [Pratap Kumar 2016], firewall [Last 2016], identificar técnicas de arquivos executáveis que estão ligados a explorações de vulnerabilidades conhecidas [Leyla Bilge 2012] e através do PatchGen em que detecta segmentos de códigos que são os mesmos com os *patches* do banco de dados e em seguida identificas os *patches* que devem ser corrigidos e ajustá-los [Tianyue Luo 2015].

3. Método

A prevenção de ataques do tipo *zero-day*, irá ocorrer pela análise da combinação de dados de vulnerabilidades que já foram divulgadas. Iremos utilizar três sites que divulgam essas informações, que são: o *National Vulnerability Database* (NVD), *National Institute of Standards and Technology* (NIST), *Common Vulnerabilities and Exposures* (CVE) e *Listing of Threats & Risk*, da Symantec.

Analisando essas três fontes de informações percebemos algumas pecularidades das informações obtidas. A *List of Threats & Risk* é a que possui mais detalhes que as demais em que temos o tipo vulnerabilidade, podendo ela ser um *worm*, vírus, *spyware*, macro, *trojan* ou *trojan horse*, etc. Além do que, nessa mesma lista conseguesse visualizar o sistema que é afetado, a data de descoberta, o que essa vulnerabilidade realiza com muitos detalhes, se arquivos são criados devido a essa falha, o risco de impacto e alguns casos como que se reproduz essa vulnerabilidade.

Com relação as informações do *National Vulnerability Database* (NVD), são identificados por meio de *Common Vulnerabilities Exposures* (CVE) que é um dicionário de nomes comuns para vulnerabilidades de segurança cibernética conhecidas publicamente [MITRE 2017]. As informações contidas tem informações relevantes e algumas que são similares da bases de dados anterior, os campos são complexidade de acesso, autenticação, tipo do impacto e também conseguimos também visualizar por

fornecedores de *software* quais falhas que ocorreram em suas aplicações. Porém, não existem todos os identificadores e os fornecdores no *National Vulnerability Database*, somente alguns estão presente. Devido a esse fato fomos atrás dos demais identificadores, entretanto um problema que achamos é que cada indetificador possuem uma breve descrição e algumas referências, sendo que algumas delas possuem links que mostram os erros causados.

Acreditamos que combinar essas informações desses três conjuntos complementam uns aos outros, e julgamos que o resultado disso pode ser uma maneira muito útil para prevenir ataques cibernéticos como o ataque *zero-day*. Por exemplo, reproduzindo esses ataques contidos nas informações extraídas podemos verificar se as aplicações tratam das vulnerabilidades, saber quais sistemas ou fornecedores de *software* possuem mais falhas do que outros, fará com o usuário opte por uma outra alternativa que não possua tantas vulnerabilidades ou ir para uma aplicação seja preferidas do cibercriminosos, já que se optar ele será mais um alvo para os criminosos. Além disso, podemos complementar a reprodução de um ataque veirificando se alguma arquivo foi criado, depois de realizado o ataque, já que podemos arquivos que executem determinadas ações dentro de uma aplicação. Enfim, também podemos averiguar se os ataques de *zero-day* que já foram divulgados e que causaram algumas prejuízos em algumas aplicações, causam danos no *software*, e caso seja positivo o fornecedor poder lançar atualizações para tentar solucionar a vulnerabilidade.

Como dito anteriormente que no *National Vulnerability Database* (NVD), temos uma quantidade menor de vulnerabilidades com relação ao *Common Vulnerabilities Exposures* (CVE), podemos utilizar os identificadores para pesquisar mais detalhes das vulnerabilidades, de como reproduzir, quais são os prejuízos causados, etc. Assim como ter as aplicações no nosso sistema atualizado na versão mais recente, ter esses dados sempre mais atualizados fará com que as aplicações fiquem menos vulneráveis, já que existem inúmeros identficadores que irão receber novas vulnerabilidades que irão ser divulgadas.

4. Resultados esperados

Os resultados que esperamos é que conseguimos prevenir os ataques *zero-day*, mas estamos cientes de que alguns ataques não conseguiram ser prevenidos, já que só iremos saber depois qual vulnerabilidade foi usada para efetuar o ataque. Além do que, esperamos que essa seja mais uma maneira de prevenção, e que ainda é válido utilizar *firewall*, sensores, teste de penetração, antivírus, atualizações de navegadores e *software* para prevenir esse ataque e os demais ataque.

Além de precaver desejamos auxiliar os fornecedores de *software* a não lançarem aplicações com vulnerabilidades e sim testá-las antes e caso necessite corrigindo antes de disponibilizar elas, e que fornecedores que tem aplicações já sendo utilizadas que testem elas e que caso haja alguma anomalia que lançem atualizações para suas aplicações evitando que seus dados ou os dados dos usuários que utilizam sejam expostos a pessoas não-autorizadas.

5. Trabalhos relatados

No artigo *Before We Knew It An Empirical Study of Zero-Day Attacks In The Real World* [Leyla Bilge 2012], foi desenvolvido uma técnica para identificar e analisar ataques de *zero-day* a partir dos dados disponíveis do *World Intelligence Network Environment* (WINE), em que identificava arquivos executáveis que estão ligados as exporações de vulnerabilidades conhecidas. Os resultados obtidos foram que 60% das vulnerabilidades que identificou ainda não eram conhecidas.

David Last em seu artigo *Forecasting Zero-Day Vulnerabilities* [Last 2016], propõe gerar previsões de descoberta de vulnerabilidades nos níveis global e categoria (navegador, sistema operacional e player de vídeo), a partir de quatro conjuntos de dados sendo elas GlobalVuns, BrowserVulns, OSVulns e VideoVulns. O mesmo define três modelos de suítes, em que a primeira seria a combinação de três modelos de previsão que seria o *Composite Regression*, modelo de regressão linear e modelo de regressão combinada. O segundo modelo de suíte seria a aprendizagem de máquinas sobre vulenrabilidades cumulativas, em que utilizou-se o Weka para analisar dados entre janeiro de 2000 até o início do período de previsão. Por fim, o último a aprendizagem de máquinas em vulnerabilidades mensasi, em que é similiar a anterior, exceto que treina e prevê dados mensais de vulnerabilidades de software em vez de dados cumulativos. Os resultados no *Composite Regression*, apresentaram desmepenhos melhores, e conjuntos de vulnerabilidades acumuláveis e mensais há um sobreposição de dados/sem sobreposição de dados. Já com as duas últimas suítes Last obteu melhores resultados para do OSVulns em modelos cumulativos de Aprendizagem de Máquinas, e BrowserVulns e VideoVulns, os melhores resultados foram obtidos nos modelos mensais Aprendizagem de Máquina.

No artigo [Pratap Kumar 2016], utilizou-se técnicas baseadas em estatísticas, assinaturas, comporamento e híbrida. O objetivo delas seriam identificar as façanhas em tempo real e tentar eliminar em tempo real, separação de ataque específico ou minimizar os danos provocados pelo ataque. Na técnica baseada em estática qualquer coisa que está acontecendo de fora do normal é bloqueada ou marcada. A segunda técnica é a baseada em assinaturas, porém requer uma assinatura para estar na biblioteca de assinaturas para a detecção. Dentro dessa técnica, temos outras três subtécnicas que seriam assinatura baseado em conteúdo, que compara o conteúdo dos pacotes com assinaturas mal-intencionadas conhecidas, assinatura baseada na semântica são caras para gerar e não podem ser implementadas, e não podem ser implementadas em sistema de detecção de intrusão e a última subtécnica é assinatura baseada em vulnerabilidades, em que testamos o ponto de vulnerabilidades, indicando uma mensagem de entrada fará com que a execução do programa atinja o ponto de vulnerabilidade. A penúltima técnica é prever o comportamento futuro de um servidor web, máquina vítima, a fim de negar quaisquer comporamentos que não são esperados. Por fim, a última técnica seria a combinação das anteriores derrubando a fraqueza em qualquer filtro de tráfego suspeito.

Em *0-Day Vulnerabilities and Cybercrime* [Paolo Foti 2015], é realizado várias entrevistas que mostram sua ideias, realtam sobre os mercados de vulnerabilidades existentes, fábricas de *malwares*, cibercrime e o cibercrime em cenário militares. No penúltimo artigo [Serge Egelman 2013] assim como no anterior relatou mais sobre os mercados de *zero-day*. Enfim, o último [Tianyue Luo 2015] é apresentado uma

ferramenta, em que detectamos segmentos de código que são os mesmo com patches no banco de dados, identificamos patches que devem ser corrigidos e ajustá-los. Com essa aplicação gerou-se rapidamente um patch de código vulnerável de um dia, e um exemplo citado é que foi gerado 1187 pachtes que provavelmente haviam vulnerabilidades reais, analisando mais de 2 bilhões de linhas de código.

Referências

Last, D. (2016). Forecasting zero-day vulnerabilities. *11th Annual Cyber and Information Security Research Conference*, 13.

Leyla Bilge, T. D., editor (2012). *Before we knew it: an empirical study of zero-day attacks in the real world*. ACM.

MITRE (2017). About CVE.

Paolo Foti, Jart Armin, M. C., editor (2015). *0-Day Vulnerabilities and Cybercrime*. IEEE.

Pratap Kumar, R. K. S. (2016). A review on 0-day vulnerability testing in web application. *2nd International Conference on Information and Communication Technology for Competitive Strategies*, 141.

Serge Egelman, Cormac Herley, P. C. v. O., editor (2013). *Markets for Zero-Day Exploits: Ethics and Implications*. ACM.

Spitzner, L. (2003). Honeypots: Simple, cost-effective detection.

Tianyue Luo, Chen Ni, Q. H. M. Y. J. W. Y. W., editor (2015). *POSTER: PatchGen: Towards Automated Patch Detection and Generation for 1-Day Vulnerabilities*. ACM.