Aplicações Web Legadas: Avaliação e Estratégias para Gerenciamento do Risco

Ricardo Almeida[1], Victor Covalski[1], Thiago Cardoso[1], Rafael Padilha[1], Thomas Oliveira[1], Roger Machado[1], Ricardo Ladeira[2]

[1]Universidade Federal de Pelotas (UFPel), Pelotas – RS – Brasil

[2]Instituto Federal Catarinense (IFC), Blumenau – SC – Brasil

{ricardo.almeida,vrcjunes,thiago.cardoso,rafael.padilha,

thomas.aguiar,rmachado.ifm}@ufpel.edu.br, ricardo.ladeira@ifc.edu.br

Resumo. *Este artigo descreve a abordagem realizada na Universidade Federal de Pelotas para gerenciamento dos riscos de segurança de aplicações web legadas. Essa abordagem consistiu de quatro etapas: (i) identificação dos sistemas legados; (ii) seleção dos sistemas com maior probabilidade de riscos; (iii) execução de ferramentas de testes de intrusão; (iv) implementação das opções de mitigação de riscos, reduzindo-os a níveis aceitáveis. Além disso, são discutidos os equívocos cometidos durante a implantação de um Web Application Firewall, considerado como uma das estratégias para mitigação. Ainda, será descrita a nova metodologia empregada após o aprendizado com os equívocos cometidos.*

1. Introdução

Muitas empresas estão focadas em vetores de ameaças emergentes. Como consequência, elas podem acabar ignorando os riscos representados por aplicações legadas. Nas organizações que continuamente criam novas aplicações, frequentemente ao longo do tempo elas deixam de ser mantidas, porém permanecem operacionais geralmente por meses ou até anos [Kulkarni 2015]. Com isso, além dos riscos já existentes, muitas vezes por não seguirem boas práticas de segurança no desenvolvimento do código, novos riscos acabam surgindo em função da descoberta de novas vulnerabilidades e devido a evolução na sofisticação dos ataques. Em alguns casos, estas aplicações consistem de sistemas estratégicos para as atividades administrativas das instituições, logo, a importância do gerenciamento dos riscos associados à aplicações legadas se potencializa.

Na UFPel, por diversos fatores, incluindo a falta de normas internas baseadas na Norma Complementar nº 16 - a qual estabelece Diretrizes para Desenvolvimento e Obtenção de Software Seguro nos Órgãos e Entidade da Administração Pública Federal - a aquisição de software acaba ocorrendo de maneira indiscriminada, sendo que, em alguns casos, o setor de Tecnologia da Informação (TI) é envolvido apenas quando os responsáveis pela compra necessitam de suporte técnico. Isto significa que grande parte dos código legados são originados fora da Universidade ou são escritos por estagiários que há muito tempo já deixaram a instituição.

De acordo com a Gartner, 73% dos novos ataques têm como foco a aplicação [AlertLogic 2017, Verizon 2018]. Adicionalmente à isto, o *National Vulnerability Database* (NVD) mantido pelo *National Institute for Standards and Technology* (NIST) aponta

que 92% das vulnerabilidades estão na aplicação [Verizon 2018]. No que diz respeito à aplicações web, *Cross-Site Scripting* e *SQL Injection* são duas das principais vulnerabilidades [MITRE 2011, Foundation 2017]. O Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br[1]) ressalta que no ano de 2017 houve um aumento de 10% em relação a 2016 nas notificações de ataques a servidores Web. Ainda segundo o CERT.br, os atacantes exploram vulnerabilidades em aplicações Web para comprometer sistemas e então realizar as mais diversas ações, tais como: hospedar páginas falsas de instituições financeiras; armazenar ferramentas utilizadas em ataques; e propagar spam e/ou scam.

Além de considerar estes dados, para definição da prioridade deste projeto, a gestão considerou os ataques previamente sofridos em aplicações web. Alguns destes consistiram de ataques simples de *defacement*, no entanto, outros impactarem consideravelmente a continuidade dos negócios da Universidade, especialmente afetando a disponibilidade dos serviços.

Tendo este cenário em vista, o objetivo central deste trabalho é apresentar as estratégias adotadas na UFPel para gerenciamento dos riscos de segurança tendo como escopo as aplicações web legadas. As etapas seguidas foram baseadas em [Weber 2013]. Serão discutidos também os equívocos realizados durante a implantação de um *Web Application Firewall* (WAF), o que propiciou a maturidade da equipe no uso desta tecnologia, bem como nos conceitos de segurança da informação de forma geral.

2. Métodos

No ambiente computacional da UFPel existiam três servidor principais utilizados para hospedagem de sites mantidos por terceiros:

- o primeiro consiste de uma instalação multisite do Wordpress, o qual é oferecido atualmente como plataforma institucional de hospedagem, possuindo normas de armazenamento, backup e atualizações bem definidas, além de uma equipe dedicada para sua manutenção;
- o servidor X, foi utilizando durante anos como solução de hospedagem de sites, porém sem possuir controles de segurança, sendo utilizado tanto por aplicações desenvolvidas por terceiros como por sistemas estratégicos para algumas atividades administrativas;
- o servidor Y representou um aprimoramento do X, sendo baseado em um painel de controle de hospedagem de sites *Free and Open Source Software* (FOSS), o qual oferece algumas facilidades no gerenciamento em conjunto com controles simples de segurança, como por exemplo, ajuste de permissões automaticamente.

O escopo de avaliação de risco teve como alvo os dois últimos servidores responsáveis por hospedar sites que não são mantidos pelo setor de TI, tendo como foco as aplicações web. Ou seja, neste momento não foram consideradas as vulnerabilidades existentes por configurações frágeis dos servidores web (opções do PHP ou do apache) e nem associadas ao sistema operacional ou infraestrutura de rede.

Inicialmente, sabendo-se das inúmeras vulnerabilidades existentes nas aplicações web hospedadas nestes servidores e considerando os ataques previamente sofridos, foram

[1] `https://www.cert.br/stats/incidentes/2017-jan-dec/analise.html`

realizados ajustes de *hardening* nos servidores para na sequência realizar uma tentativa de implantação do WAF ModSecurity[2]. De forma resumida, ele foi instalado em um servidor dedicado operando como um proxy reverso para o servidor X, sendo configurado com as regras da OWASP[3].

Após a instalação, ele foi colocado para operar em modo de detecção ("SecRuleEngine DetectionOnly"), conforme recomendado nas perguntas frequentes do ModSecurity[4]. Porém, o ModSecurity foi ativado ("SecRuleEngine On") logo em seguida, sem a devida análise aprofundada dos eventos gerados, uma vez que naquele momento a UFPel ainda não possuía uma solução de gerenciamento de logs para auxiliar esta tarefa. Esta ação equivocada gerou muitos falsos positivos, especialmente nos painéis de administração dos sites presentes no servidor X (`/admin`, `/wp-admin`, entre outros).

A segunda abordagem realizada consistiu primeiramente em manter o ModSecurity operando como proxy reverso para o servidor X em modo de detecção. Esta decisão, além de gerar eventos importantes em caso de algum comprometimento, foi útil para análise detalhada e ajustes finos no segundo momento, quando ocorreu a ativação por site. Simultaneamente à este projeto, foi realizada a implantação da solução Elastic Stack, a qual foi importante para auxílio na análise do eventos produzidos pelo ModSecurity.

Na sequência, ocorreu a realização de análise da data de modificação dos arquivos existentes nos diferentes sites hospedados no servidor X em conjunto com uma análise visual do site. Desta forma, os sites que estavam desatualizados há mais tempo foram elencados primeiramente para contato com os responsáveis na tentativa de desativação. Nos casos em que houve a permissão para desativação, não foi necessária a execução do teste de intrusão.

Posteriormente, foram realizados testes de intrusão simples tanto para os sites em que não houve retorno inicial como para os demais sites. As ferramentas sqlmap[5] e w3af[6] foram utilizadas. Ressalta-se a importância da geração e arquivamento de forma organizada dos relatórios produzidos por estas ferramentas. Uma vez que consistiam de relatórios técnicos, foi considerado inapropriado o envio por e-mail, com isso, apenas em casos de maior resistência, quando se fez necessário reuniões, eles foram utilizados para respaldo na justificativa para desativação do site.

Tanto para os sites considerados mais antigos, como para os que apresentaram vulnerabilidades nos testes de intrusão, houve uma primeira tentativa de contato através de e-mail com os responsáveis cadastrados em base mantida pela equipe de "web sites". Em alguns casos, também foram considerados os contatos existentes nos próprios sites. O e-mail enviado informava que o serviço de hospedagem destes sites desenvolvidos por terceiros estava sendo descontinuado, possibilitando a migração do site para a plataforma do Wordpress Institucional, sendo destacadas as seguintes vantagens:

- Não possui custo adicional para a instituição;

[2] `https://www.modsecurity.org`
[3] `https://github.com/SpiderLabs/owasp-modsecurity-crs`
[4] `https://github.com/SpiderLabs/ModSecurity/wiki/ModSecurity-Frequently-Asked-Questions-(FAQ)`
[5] `http://sqlmap.org`
[6] `http://w3af.org`

- A plataforma Wordpress possui uma grande comunidade de desenvolvedores, logo, sua interface, funcionalidades, desempenho e segurança estão em constante atualização;
- Não utiliza códigos descontinuados que possam aumentar as possíveis falhas de segurança exploradas para fins maliciosos;
- Fácil manutenção do site, pois o Wordpress possui uma interface simples para postagem de conteúdos;
- O Wordpress Institucional é atualizado conforme as correções da própria ferramenta são anunciadas;
- Adequação à identidade visual da UFPel.

Caso os responsáveis não retornassem em um prazo de 7 dias, o modsecurity era ativado considerando a desativação de algumas regras que tivessem gerado falsos positivos nos acessos prévios e em acesso realizado pelo responsável por executar a ativação. Além disso, outro e-mail era enviado informando da desativação do site em 30 dias caso não houvesse um retorno.

Naturalmente, houveram alguns casos de resistência, sendo fundamental o apoio da alta gestão. Os gestores, junto a equipe de segurança que realizou os testes de intrusão, conversaram pessoalmente com os responsáveis pelos sites. Nas reuniões eram informadas as vulnerabilidades identificadas e os riscos envolvidos, flexibilizando o tempo de manutenção do site vulnerável em operação, desde que o ModSecurity se mantivesse ativo até a realização da migração.

Em casos especias, não houve a possibilidade de desativação e nem de migração do site, com isso, o ModSecurity em conjunto com o acesso restrito ao painel de administração, foram fundamentais para minimizar o risco de maneira efetiva e permanente, uma vez que os novos testes de intrusão não acusaram as vulnerabilidades previamente identificadas.

Alguns casos críticos ocorreram, principalmente, com alguns sistemas estratégicos empregados nas atividades administrativas da UFPel, onde a ativação do ModSecurity foi inviável considerando a alta taxa de falsos positivos em função da forma de desenvolvimento dessas aplicações. Com isso, o controle adotado foi de restringir o acesso para apenas oriundo da rede interna da UFPel ou do serviço de proxy, desta forma flexibilizando o acesso externo. Em paralelo, alguns desses sistemas estão sendo desenvolvidos pela equipe de desenvolvimento de software, de maneira integrada com o sistema integrado de gestão produzido na UFPel.

O mesmo processo foi realizado para o servidor Y, com a única diferença que neste foi realizada a implantação do ModSecurity no próprio servidor. Esta decisão visou propiciar a integração com o painel de controle de hospedagem utilizado neste, e evitar um ponto único de falha que ocorreria caso fosse optado por utilizar a mesma implantação do ModSecurity como proxy reverso.

3. Resultados

Como resultado, é possível destacar que, dos sites avaliados, apenas nos que foram desenvolvidos em HTML ou que não realizavam a passagem de parâmetros, não foram encontradas vulnerabilidades. Como consequência, após aproximadamente 2 anos de

execução deste projeto, cerca de 20 sites foram desativados, 30 migrados e 20 tiveram o ModSecurity ativado e o acesso ao painel de administração restringido, não sendo novamente encontradas as vulnerabilidades anteriormente identificadas com a execução das ferramentas mencionadas para testes de intrusão. Além destes, 5 sites tiveram seu acesso exclusivamente restringido apenas para a rede interna da UFPel.

Evidencia-se também que, até o momento da escrita deste documento, não houveram mais incidentes associados às aplicações web protegidas pela abordagem. Além disso, não foram registradas reclamações relativas a falsos positivos. Contudo, observa-se que o projeto ainda está em continuidade.

4. Conclusão

A abordagem adotada, além de oferecer maior proteção ao ambiente computacional, propiciou o amadurecimento da equipe em questões de vulnerabilidades web e implantações de WAF. Apesar de ainda haverem sites desprotegidos nos servidores mencionados, o uso do ModSecurity no modo de detecção produz evidências importantes para possíveis investigações de incidentes.

Ressalta-se que o processo está em andamento e em constante evolução, existindo ainda diferentes vulnerabilidades associadas aos serviços web. Concomitantemente a este projeto, a implantação do Elastic Stack para gerenciamento de logs foi estendida para contemplar a integração com o OpenVAS[7] para análise de vulnerabilidades e com o CorReactive[8] que permite a correlação de eventos e reação a atividades suspeitas detectadas através de regras escritas com sintaxe similar à SQL.

Como trabalhos futuros, além da continuidade do projeto, pretende-se consolidar as ferramentas de análise de vulnerabilidades e de testes de intrusão, estabelecendo uma melhor automatização. Na sequência, será realizada a integração dessas ferramentas com solução para tratamento automatizado de incidentes de segurança.

Referências

AlertLogic (2017). Cloud security report. Technical report, Alert Logic.

Foundation, O. (2017). Owasp top ten project. Publication, OWASP Foundation.

Kulkarni, S. (2015). acesso em: 12 abr 2018. OWASP - OWASP Supporting Legacy Web Applications in the Current Environment Project. Disponível em: <https://www.owasp.org/index.php/OWASP_Supporting_Legacy_Web_Applications_in_the_Current_Environment_Project>.

MITRE (2011). Mitre - cwe/sans top 25 most dangerous software errors. Publication, MITRE.

Verizon (2018). Data breach investigations report. Technical report, Verizon Enterprise.

Weber, C. C. (2013). acesso em: 12 abr 2018. CERT-US - Assessing Security Risk In Legacy Systems. Disponível em: <https://www.us-cert.gov/bsi/articles/best-practices/legacy-systems/assessing-security-risk-in-legacy-systems>.

[7] http://www.openvas.org

[8] https://sourceforge.net/p/correactive/wiki/FAQ/