Indicadores de Segurança em Plataformas de Monitorização

Raimundo Chipongue, Hugo Miranda, and António Broega

Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa
fc48807@alunos.ciencias.ulisboa.pt,{hamiranda,ajbroega}@ciencias.ulisboa.pt

Resumo A separação entre administradores de sistemas e redes (TI) e especialistas de segurança afeta a capacidade e tempo de resposta a incidentes de muitas organizações. Esta separação é também observável na limitada oferta de módulos relacionados com segurança nos repositórios públicos de plataformas de monitorização. No entanto, a integração da visão de administração de TI com a da equipa de segurança, por exemplo nas consolas das plataformas de monitorização, pode contribuir para um diagnóstico mais eficaz e para a deteção precoce e resolução de incidentes. Este artigo discute a contribuição das ferramentas de monitorização no colmatar desta separação e propõe um conjunto de módulos para a plataforma Nagios, que procuram proativamente vulnerabilidades conhecidas e comportamentos anómalos, indicadores de potenciais problemas de segurança. A avaliação dos módulos em ambiente de produção e a sua aprovação pela comunidade de editores do repositório oficial de módulos do Nagios fazem-nos concluir que é possível usar a flexibilidade destas ferramentas para monitorização conjunta de sistemas, redes e eventos de segurança, facilitando e acelerando a sua deteção e resolução.

Palavras-chave: Monitorização, Segurança, Nagios.

1 Introdução

A segurança é apenas mais um dos elementos que pode afetar a disponibilidade e desempenho de toda uma infraestrutura de sistemas de informação de uma organização. Em redes corporativas com um número considerável de equipamentos, a tarefa dos administradores de tecnologias de informação (TI) deixa de ser trivial. Os sistemas de monitorização são uma ferramenta importante na deteção de comportamentos fora dos padrões previamente estabelecidos, gerando notificações em tempo real, contribuindo desta forma para acelerar a deteção e resolução de problemas.

Culturalmente, existe uma separação entre administradores de TI e técnicos de segurança ou, nalguns casos, a um isolamento das funções dos segundos apenas numa das áreas cobertas pelos primeiros. Esta separação, apesar de compreensível do ponto de vista organizacional, não é de todo benéfica pois pode levar a um atraso no diagnóstico ou mesmo a que um incidente tenha duas interpretações contraditórias, possivelmente levando à tomada de ações contraproducentes. Por

exemplo, apesar de se tratar de uma ação maliciosa, a ocorrência de um ataque de negação de serviço pode ser interpretado inicialmente como uma limitação de desempenho ou perda de disponibilidade pelo administrador do sistema.

1.1 Motivação

As plataformas de monitorização de TI desempenham tipicamente um duplo papel. Por um lado, atuam como agente profilático, permitindo detetar antecipadamente problemas que se não forem corrigidos atempadamente poderão resultar em incidentes sérios. Por outro, facilitam as operações de diagnóstico de incidentes, apresentando conjuntos de sintomas que se corretamente interpretados pelos administradores de TI aceleram a sua resolução.

A maioria das plataformas de monitorização não favorece a integração das perspetivas dos administradores de TI e dos especialistas de segurança informática. Em repositórios de algumas das mais populares plataformas de monitorização, como por exemplo, o Nagios Exchange,[1] os indicadores de segurança são abordados de forma relativamente marginal. Este estudo contribui para aproximar as duas perspetivas, ao encorajar a que os administradores de TI e especialistas de segurança tenham uma visão única do sistema através da plataforma de monitorização Nagios. O objetivo é por um lado, alargar o carácter profilático das ferramentas de monitorização aos sintomas evidenciados por falhas de segurança e por outro facilitar uma análise cruzada da informação de diagnóstico, diminuindo o impacto dos incidentes.

Este trabalho é motivado por casos como o recente ciberataque à escala global com Ransomware *WannaCry* [1], ocorrido em Maio de 2017. Apesar de explorar uma vulnerabilidade entretanto corrigida nas atualizações do sistema operativo Windows o ataque teve um impacto considerável que poderia ter sido mitigado em sistemas com plataformas de monitorização a verificarem as atualizações aplicadas aos computadores pessoais.

O artigo apresenta uma contribuição para a comunidade de utilizadores da plataforma de monitorização Nagios, descrevendo um conjunto de módulos que permitem sinalizar vulnerabilidades de segurança nos sistemas de uma organização. Estes módulos foram recentemente submetidos e aprovados pelos editores do Nagios Exchange encontrando-se publicamente disponíveis nesse repositório.

2 Trabalho Relacionado

Existe uma série de projetos disponíveis que abordam as contribuições das plataformas de monitorização nos processos de administração de uma infraestrutura de TI. Muitos destes projetos cingem-se ao estudo e utilização das plataformas como ferramenta para auxiliar a administração [22,18,2] ou apontando os pontos fortes e fracos [19]. O trabalho apresentado em [12] distingue-se dos restantes por se focar numa comparação de diferentes aproximações.

[1] `https://exchange.nagios.org`

É cada vez mais comum para grandes empresas o uso de ferramentas de Gestão de Eventos e Informação de Segurança (*Security Information and Event Management*, SIEM) para auxiliar a monitorização segura de infra-estruturas de redes e sistemas. Os SIEM são importantes componentes em TI, capazes de recolher, processar, analisar, armazenar e correlacionar registos (logs) ou fontes de informação de eventos de segurança, produzidos por ferramentas como Sistemas de Deteção de Intrusões (*Intrusion Detection Systems*, IDS), Anteparas de Segurança (*Firewalls*), Gestores de Identidades de Acesso (*Identity Access Manager*, IAM) [9], entre outros. São usados para ajudar a responder rapidamente a ataques e organizar dados de log. Entretanto a implementação e manutenção de um SIEM é um processo bastante complexo e com elevados custos, o que torna inviável para maioria das pequenas e médias empresas.

Por outro lado, apesar de alguns sistemas de monitorização como o Nagios, poderem ser utilizados como fontes de informação dos SIEM, a sua correta configuração, aliada ao desenvolvimento de módulos específicos transforma estas plataformas em alternativas viáveis e de baixo custo aos SIEM's.

Apesar da variedade de trabalhos publicados, não é fácil encontrar trabalhos que partilhem o foco deste artigo. Ou seja, que abordem as plataformas de monitorização, e que tirem proveito das suas potencialidades para gerar indicadores de segurança e consequentemente incrementar a segurança dos sistemas.

2.1 Nagios

O Nagios[2] [6] é uma popular ferramenta de monitorização, distribuída ao abrigo da licença GPL. Usa um núcleo que recorre a plugins para executar tarefas específicas, exibindo os resultados numa interface Web. Além disso, dispõe de um serviço de notificações modular, ao qual podem ser associadas ferramentas de envio de correio eletrónico, SMS, ou qualquer outro meio definido. O acompanhamento exaustivo, administração centralizada, notificações em tempo real, correção automática de problemas (nalguns casos), disponibilização de relatórios, estatísticas e fácil integração, são alguns de seus benefícios.

Plugins Os módulos ou plugins são pequenos programas externos [5] responsáveis pela realização das operações de diagnóstico e pela interpretação dos resultados. A principal informação entregue por um módulo ao núcleo do Nagios é um de quatro possíveis estados da entidade monitorizada [20]:

OK quando nenhuma anomalia for detetada;
WARNING estado intermédio espectávelmente apresentado quando o indicador ainda não atingiu o limite normal de funcionamento;
CRITICAL quando o indicador exceder o limite máximo de funcionamento ou não é recebida resposta;
UNKNOWN quando ocorre um erro na execução do módulo, incapacidade de obter informação por razões conhecidas ou incorreção na definição de parâmetros.

[2] `https://www.nagios.org/`

Agentes Para suportar a execução remota de operações de monitorização, o Nagios recorre a agentes, instalados no sistema remoto e que em resposta a uma invocação executam o plugin designado pela plataforma. A comunicação entre a plataforma pode ser realizada utilizando qualquer protocolo, sendo os principais o standard *Simple Network Management Protocol* (SNMP) [7,18], e o *Nagios Remote Plugin Executor* (NRPE), cocretizado pelo agente com o mesmo nome para o sistema operativo Linux e pelos agentes *Nagios Service Check Acceptor* (NSCA) e *NSClient++* para SOs Windows [5,18,21].

3 Módulos Desenvolvidos

Os módulos desenvolvidos têm a capacidade de monitorizar sistemas e serviços e produzir alertas de segurança através da pesquisa e análise de estados que sinalizem a subversão das políticas de segurança dos sistemas.

A seleção de módulos teve por base um levantamento de requisitos junto da equipa de administração de redes/sistemas e segurança da infraestrutura de TI duma organização de ensino com mais de 5000 utilizadores e com uma grande variedade de requisitos, incluindo por exemplo a criação de web sites específicos, a criação de redes próprias e o alojamento de equipamentos de centro de dados. Esta realidade permitiu que o processo de idealização, desenvolvimento e testes tivesse o acompanhamento da equipa de administração, dando feedback sobre as opções tomadas, sugerindo alterações, realizando testes e validando as opções tomadas. No principio deste processo foram identificados os procedimentos mais comuns e as vulnerabilidades cuja verificação estava já sistematizada mas era despoletada e realizada com procedimentos que obrigavam à intervenção da equipa de administração.

Os módulos foram desenvolvidos usando a linguagem de programação Python, e recorrendo a utilitários Linux e ferramentas de código aberto.

Monitorização do Ficheiro de Log do Servidor Apache O Apache[3] é um servidor Web bastante popular, usado principalmente em sistemas operativos Linux. Ao responder às solicitações, este servidor regista num ficheiro de *log* o código de resposta HTTP devolvido ao cliente [14]. Os códigos HTTP estão agrupados em 5 classes conhecidas como 1xx, 2xx, 3xx, 4xx e 5xx.

Os códigos da classe 4xx sinalizam erros nos pedidos por parte do cliente. Em particular os códigos 401, 403 e 404 sinalizam respetivamente a tentativa de acesso a conteúdo não autorizado, conteúdo para o qual não foram apresentadas credenciais de autenticação válidas ou tentativa de acesso a conteúdo inexistente. Registos repetidos de acessos que gerem estes códigos sugerem uma provável tentativa de subversão das políticas de segurança por parte de um ou um conjunto de clientes que se encontrem a averiguar a existência de vulnerabilidades na configuração do servidor.

[3] `http://www.apache.org/`

O módulo Nagios ***check_apache_status*** foi desenvolvido para auditar o ficheiro de log do servidor Apache, procurando ocorrências dos códigos 401, 403 e 404. Em caso de deteção são gerados alertas com os estados *WARNING* ou *CRITICAL*, dependendo da parametrização definida que contabiliza a quantidade máxima aceitável de códigos no ficheiro de *log* e por endereço IP. O módulo possibilita ainda a definição da quantidade de registos a auditar.

Monitorização dos Programas Instalados O controlo do conjunto de ficheiros executáveis instalados nas plataformas Linux é uma tarefa complexa devido à sua grande quantidade e distribuição por um conjunto de diretórios. Adicionalmente, a quantidade de programas pode variar a cada instalação ou atualização de pacotes de software ou do sistema operativo. Esta arquitetura torna os sistemas Linux vulneráveis à instalação de troianos ou *backdoors* que podem passar facilmente despercebidos a um administrador de TI.

O módulo ***check_app*** auxilia a equipa de administração por monitorizar a lista de executáveis instalados num conjunto de diretorias predefinidas e configuráveis pelo utilizador. O levantamento é realizado a cada execução e comparado com a listagem obtida na execução anterior, sinalizando anomalias com o estado *CRITICAL*.

Monitorização de Ataques TCP SYN Flood A inundação por pedidos de estabelecimento de ligação (*SYN Flood Attack*) é um tipo de ataque da categoria *Denial of Service* (DoS) ou *Distributed Denial of Service* (DDoS) que consiste na sobrecarga direta da camada de transporte do alvo e indireta na camada de aplicação do modelo OSI. É concretizado pelo envio de sequências de pedidos de estabelecimento de ligação do protocolo TCP (SYN) [10,15].

A deteção destes eventos é feita usando o módulo ***check_synflood***, que monitoriza as ligações de rede em busca de um número de ligações com o estado SYN_RECV acima dos limites definidos pelo utilizador. Consoante os limiares definidos, o módulo pode retornar o estado *WARNING* ou *CRITICAL*.

Monitorização do Conteúdo de Paginas Web A alteração maliciosa de conteúdo disponibilizados através da World Wide Web (WWW) é um ataque bastante usado por *hackers* e vulgarmente designado por *defacement* [13]. O *defacement* tem consequências sobretudo para a imagem da organização, ao expôr a sua vulnerabilidade ou anunciando informação contraditória com os seus objetivos. Tecnicamente, este ataque consubstancia-se na subversão de duas das principais propriedades de segurança da informação, nomeadamente a integridade e a autenticidade.

O módulo ***check_defacement*** monitoriza o conteúdo de páginas web recebidas como argumento. Este módulo verifica se o conteúdo da página apresenta palavras tipicamente utilizadas pelos agentes maliciosos nos ataques de *defacement* sinalizando a sua ocorrência com o estado *CRITICAL*. Por omissão foi definido um conjunto destas palavras, que pode ser alterado através da edição do código ou pela utilização de argumentos na linha de comando.

Monitorização da Correspondência Entre Nomes de Domínio e Endereço IP Associado Um nome de domínio é uma referência de fácil memorização associada a um endereço IP e que serve para identificar um computador na Internet. As informações de correspondência entre nomes de domínios e endereços IP são armazenadas em base de dados de servidores *Domain Name System* (DNS), que asseguram a indicação do endereço certo para a entrega dos pedidos. No entanto, este sistema não é imune a ataques tais como ***DNS cache poisoning***, caraterizado por comprometer a integridade do servidor de DNS. Este ataque é concretizado através da injeção de informações falsas, deturpando a precisão das consultas DNS. Na prática, estes ataques permitem o redirecionamento do tráfego, permitindo a personificação de um endereço legítimo por terceiros [23].

O módulo ***check_dns*** compara a associação entre nomes de domínio e endereços IP em servidores de DNS externos, verificando se a resposta está de acordo com o esperado. O servidor a testar é passado como argumento utilizando-se por omissão o servidor de DNS da Google por ser um dos servidores mais populares. A deteção de incoerências é sinalizada com o estado *CRITICAL*.

Monitorização de Listas Negras As listas negras [16] são uma das mais populares ferramentas de combate ao crescente problema de *spam* e tentativas de *phishing* utilizando o correio eletrónico. As listas são normalmente implementadas sob a forma de registos DNS em domínios geridos por organizações responsáveis pela gestão das listas. Neste modelo, um servidor de correio eletrónico é inserido numa lista negra adicionando o seu endereço IP de forma invertida, por exemplo o endereço 192.168.1.1 é inserido e armazenado como 1.1.168.192.

Para verificar se um determinado endereço IP é confiável, as ferramentas *antispam* efetuam consultas ao DNS, pesquisando o endereço IP do servidor no formato invertido. O conteúdo do registo retornado é irrelevante uma vez que é a sua existência que sinaliza a presença do servidor na lista negra. Estas listas são geridas por entidades privadas, e as políticas de inserção de endereços são suscetíveis a falhas e interpretações erróneas do comportamento por vezes legítimo dos servidores de correio eletrónico.

Tipicamente, mensagens enviadas por servidores em listas negras são liminarmente rejeitadas ou marcadas pelas ferramentas de deteção de SPAM. A presença de um servidor de correio eletrónico legítimo numa lista negra tem por isso consequências negativas no funcionamento da instituição, importa por isso tomar conhecimento e resolver adições incorretas com a maior brevidade possível.

O módulo ***check_dnsbl*** facilita a deteção destes incidentes, ao reproduzir o comportamento de múltiplos filtros *antispam*, verificando nas listas negras a presença do endereço IP recebido como argumento. Por omissão, o módulo pesquisa em 27 das listas negras mais populares. A presença do servidor em pelo menos uma destas listas é assinalada com o estado *CRITICAL*.

Monitorização das Configurações DNSSEC A extensão de segurança de nomes de domínios (*DNSSEC*) usa criptografia assimétrica para assegurar a autenticidade e integridade dos dados manipulados pelo protocolo DNS. O uso do DNSSEC melhora a fiabilidade do sistema, previne ataques *(man-in-the middle, MITM)*, corrige fragilidades do protocolo DNS e diminui a probabilidade de manipulação ilícita da informação [3]. No entanto, quando mal configurado ou com assinaturas expiradas, o DNSSEC dá uma falsa sensação de proteção que pode resultar na exposição não intencional do tráfego.

O módulo ***check_dnssec*** monitoriza o servidor DNS em busca de vulnerabilidades nas configurações do DNSSEC e assinala com o estado *CRITICAL* a sua deteção. Quer o domínio a verificar quer o servidor de DNS utilizado são passados como argumento. Por omissão a validade da informação é realizada no servidor DNS da google (*8.8.8.8*) por ser um dos servidores mais populares.

Monitorização de Ficheiros e/ou Diretorias Num sistema em operação existe um conjunto de ficheiros, por exemplo de configuração, que se espera que se mantenham inalterados por longos períodos de tempo. A verificação da sua alteração pode sinalizar problemas de segurança. No entanto, este processo é particularmente moroso e sujeito a erros pelo que é altamente recomendável optar pela sua monitorização automática.

O módulo ***check_filechange*** alerta para a ocorrência de eventos de alteração do estado de ficheiros e diretorias. O módulo utiliza a ferramenta de monitorização de estados *Inotify-tools* [17], delegando desta forma no próprio sistema operativo o processo de monitorização, contribuindo para a diminuição do consumo de recursos associado a esta tarefa. O módulo recebe como argumentos os ficheiros/diretorias a serem monitorizados, os eventos a sinalizar e o caminho do ficheiro de log onde o *Inotify* registará as deteções. A ocorrência de um ou mais eventos é notificada com o estado *CRITICAL*.

Monitorização de Vulnerabilidades Web Uma vulnerabilidade é um defeito que pode ser explorado por um atacante com o objetivo de subverter a política de segurança [8]. No caso particular de servidores da World Wide Web (WWW) as vulnerabilidades mais comuns devem-se à falta de tratamento pelo serviço de partes do conteúdo do pedido, *cookies* mal configurados, identificadores de sessão expostos no navegador, parâmetros passados no *URL* ou em parâmetros de formulários, por exemplo de *login*.

Um analisador de vulnerabilidades web é uma ferramenta que percorre todo um site WWW tentando identificar pontos onde existam vulnerabilidades bem conhecidas. O módulo ***check_nikto*** usa o *analisador de vulnerabilidades web* NIKTO[4] para realizar estas auditorias. O NIKTO produz relatórios no formato HTML, alertando para a existência de boa parte das vulnerabilidades web mais populares [11]. O site monitorizado é passado como argumento, e opcionalmente podem ser especificados o porto de comunicação, a diretoria onde será armazenado o relatório, o nome e tempo de validade do relatório, assim como os tipos

[4] `https://cirt.net/nikto2`

de vulnerabilidades a serem pesquisadas. O módulo analisa o relatório produzido pelo NIKTO em busca de indicações de vulnerabilidades encontradas e sinaliza a sua deteção com o estado *CRITICAL*.

Monitorização de Portos de Comunicação Porto de comunicação é o número que identifica uma aplicação à qual se destinam os dados de uma comunicação. Desta maneira, os dados são enviados diretamente pelo sistema operativo para a aplicação correspondente. Uma vez que estes são os pontos de entrada e saída de informação na rede, portos abertos fora do controlo dos administradores podem representar uma janela para que utilizadores maliciosos possam aceder ao sistema.

O módulo ***check_open_port*** monitoriza os portos de comunicação e alerta para a abertura de portos que não façam parte da lista de portos autorizados passada como argumento. A identificação destes casos é sinalizada com o estado *CRITICAL*.

Monitorização das Configurações SSL/TLS O *Secure Sockets Layer* (SSL) e o seu sucessor *Transport Layer Security* (TLS) são protocolos criptográficos projetados para garantir autenticidade, confidencialidade e integridade na troca de dados em aplicações cliente/servidor. Estes protocolos usam certificados digitais para cifrar a informação trocada entre aplicações os quais exigem cuidados na sua administração e utilização. Por exemplo, os certificados expiram, podem ser revogados, usar protocolos criptográficos inseguros ou estar mal configurados. O comprometimento de certificados é uma situação de risco elevado por oferecer uma falsa sensação de segurança às partes em comunicação.

O módulo ***check_ssl***, monitoriza as configurações e o estado dos certificados utilizando a API online do *SSLLAB*[5]. Esta API realiza testes às configurações de certificados em busca de vulnerabilidades devolvendo uma classificação numa escala de letras e que indica o nível de gravidade das vulnerabilidades encontradas. A sinalização utiliza os estado *WARNING* ou *CRITICAL* em função da classificação atribuída, dos parâmetros estabelecidos na linha de comando e no prazo de validade do certificado.

Monitorização de Atualizações de CMS Os Content Management Systems (CMS) são ferramentas de produção e gestão de conteúdos web muito populares por oferecem interfaces particularmente amigáveis. A sua popularidade e disseminação tornam-nas também um alvo apetecível de ataques, sendo frequente a identificação de novas vulnerabilidades. O WordPress[6] é um dos CMS mais populares, desenvolvido por uma comunidade aberta, e em constante aperfeiçoamento. Esta comunidade publica frequentemente atualizações que corrigem as vulnerabilidades entretanto identificadas. No entanto, do lado do utilizador a descoberta de novas atualizações nem sempre é um processo trivial, o que leva a

[5] `https://www.ssllabs.com/`
[6] `https://wordpress.org/`

que muitas vezes se utilizem por largos períodos de tempo versões desatualizadas e com vulnerabilidades conhecidas publicamente.

O módulo ***check_wp_update*** auxilia a descoberta de atualizações de segurança deste CMS. Em contraste com outros módulos equivalentes já disponíveis o ***check_wp_update*** utiliza o sistema de ficheiros e não um pedido HTTP para aceder ao ficheiro version.php onde está registada a versão em execução. Apesar de por isso ter que ser executado localmente no servidor, esta versão tem a vantagem de não obrigar à exposição pública da informação da versão em uso e que pode ser utilizada por potenciais atacantes para identificar as vulnerabilidades conhecidas na versão em causa.

Os dados da versão mais recente são obtidos por consulta ao site do WordPress através da API disponibilizada pela equipa de desenvolvimento. A deteção de nova atualização é sinalizada com o estado *CRITICAL*, uma vez que as atualizações mais frequentes deste CMS são as conhecidas por *patchs*, disponibilizadas justamente para correções de *bugs*. Entretanto, apesar do incremento das versões *Major* e *Minor* representar essencialmente a introdução de melhorias nas funcionalidades do software, nada impede que sejam igualmente introduzidas correções de *bugs*.

4 Avaliação

Os testes foram realizados em várias plataformas Linux, tais como *Xubuntu-16-04.01, Lubuntu 16.10, Gentoo 2.2, CentOS 7 e Debian 8*.

No âmbito da avaliação foram simulados diversos cenários que permitiram testar todos os possíveis estados retornados por cada um dos módulos, validando o seu correto funcionamento.

Por exemplo os testes ao módulo *check_syn_flood* foram concretizados usando a ferramenta de ataque DoS/DDoS ***hping3***, através do comando `# hping3 <alvo> -S -p 80 --flood --rand-source`. Enquanto que os testes ao módulo *check_wp_update* passaram por usar versões desatualizadas do CMS WordPress, através do *downgrade* do software, forçando assim a geração de alertas.

Na sequencia dos testes e validação dos módulos por parte dos autores e da equipa de administradores da organização, os módulos passaram a fazer parte da biblioteca utilizada por uma instalação do Nagios a monitorizar 180 servidores (dos quais 160 virtuais) que alojam vários serviços, onde se destacam servidores *World Wide Web* (WWW), correio eletrónico, base de dados, cópias de segurança, *File Transfer Protocol* (FTP), servidores de *Webdav*, *Domain Name System* (DNS), gestão documental e gestão de acessos.

Em ambiente de produção, os módulos encontram-se instalados em maquinas com o sistema operativo Linux CentOS 7, e nos primeiros dias logo após a sua instalação contribuíram para a deteção de alguns problemas:

check_wp_update Este módulo foi o primeiro a corresponder às expetativas, ao alertar para a disponibilização da versão 4.7.4 do WordPress e posteriormente para a versão 4.8, o que permitiu que correções de segurança fossem

instaladas pouco tempo depois de disponibilizadas, reduzindo desta forma a janela de vulnerabilidades para um grande conjunto de web sites alojados no CMS.

check_nikto Este módulo também deixou boas indicações ao detetar vulnerabilidades em alguns dos sítios monitorizados. Destas vulnerabilidades, importa destacar a do tipo *Clickjacking*, que embora não seja considerada vulnerabilidade de grande criticidade, foi prontamente corrigida. Vulnerabilidades deste tipo existem quando é possível um atacante usar camadas transparentes para enganar um utilizador, fazendo com que este clique em um objeto (link ou botão) invisível. Este click pode ser utilizado para a realização de ações maliciosas [4].

check_app, check_synflood, check_filechange e check_open_port Estes módulos estão instalados e a monitorizar parte dos servidores administrados pela organização, não tendo até ao momento detetado qualquer situação anormal.

check_apache_status Este módulo foi igualmente instalado e encontra-se a monitorizar cerca de 32 servidores, sem qualquer deteção até ao momento da escrita deste artigo.

check_defacement Com características reativas encontra-se igualmente instalado e a monitorizar o principal sítio da organização, e embora não tenha até ao momento detetado qualquer situação anormal, foi considerado pela equipa de administração como um dos mais úteis.

check_dns, check_dnssec e check_ssl Estes módulos estão a monitorizar os 7 domínios mais relevantes da organização, não tendo até ao momento detetado qualquer situação anormal.

check_dnsbl Este módulo encontra-se instalado e a monitorizar 3 servidores responsáveis pelo envio de correio eletrónico, tendo por 2 vezes detetado a inclusão de um destes servidores numa lista negra, o que permitiu que se corrigi-se antes que problemas no envio de correios eletrónicos fossem detetados e reportados.

Após submissão ao repositório oficial do Nagios,[7] os módulos foram avaliados, aprovados e disponibilizados publicamente pela equipa de administração deste repositório. Encontram-se por isso disponíveis ao público dentro dos padrões de licenciamento definidos. Os módulos estão também publicamente acessíveis no repositório público github.[8]

5 Conclusões e Trabalho Futuro

Num sistema, as ferramentas de monitorização supervisionam as operações do potencialmente grande número de equipamentos conectados entre si e a Internet. Uma supervisão eficiente e eficaz é fundamental para garantir um serviço

[7] `https://exchange.nagios.org/directory/Plugins/Security/`
[8] `https://github.com/rc48807`

confiável e de alta qualidade. Neste artigo estudamos o Nagios como uma ferramenta de monitorização de código aberto com potencial para melhorar a segurança dos sistemas informáticos através do desenvolvimento de módulos focados na geração de indicadores de segurança.

O trabalho consistiu no desenvolvimento de módulos que verificam o respeito de boas práticas de segurança mas cuja verificação manual é sujeita a erros ou é repetitiva. Estes módulos foram exaustivamente testados, produzindo resultados que indicam ser possível incrementar a segurança das infraestruturas tecnológicas recorrendo à utilização das ferramentas de monitorização.

Os módulos apresentados permitem um acompanhamento automatizado, centralizado e em tempo real de eventos de segurança. Espera-se desta forma contribuir para o aproximar das equipas de administração de TI e de segurança através da partilha de uma plataforma única de monitorização e diagnóstico que ofereça uma visão conjunta e unificada destas duas realidades. Esta combinação contribuiu também para uma colaboração mais eficaz entre estas duas equipas, ambas fundamentais para o correto funcionamento dos serviços. Espera-se um incremento da pro-atividade através da reação atempada a problemas de segurança e uma redução no tempo de resposta a incidentes.

A identificação de requisitos e o desenvolvimento de novos módulos continua. Neste artigo os autores limitam-se em demonstrar e explorar a capacidade das ferramentas de monitorização na melhoria da segurança dos sistemas de acordo com os requisitos identificados por uma organização. Espera-se que num futuro próximo este estudo venha a ter continuação, através do desenvolvimento de novos módulos que dêm continuidade aos princípios que motivaram este trabalho.

Agradecimentos

O trabalho descrito neste artigo foi parcialmente suportado pela Fundação para a Ciência e Tecnologia (FCT) através do financiamento à unidade de investigação LaSIGE (ref. UID/CEC/00408/2013).

Referências

1. CERT-UE Advisory Advisory. Wannacry ransomware campaign exploiting smb vulnerability. Date: May 16 2017.
2. Abhishek Amralkar, Mayank Gaikwad, Rohit Nerkar, Pulkit Gupta, and Mukesh Waghadhare. Monitoring tools. *TechWatch Report*, pages 88–93, 2016.
3. Suranjith Ariyapperuma and Chris J Mitchell. Security vulnerabilities in dns and dnssec. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 335–342. IEEE, 2007.
4. Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. A solution for the automated detection of clickjacking attacks. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 135–144. ACM, 2010.
5. Wolfgang Barth. *Nagios: System and Network Monitoring*. No Starch Press, San Francisco, CA, USA, 2nd edition, 2008.

6. Nagios Brazilian Community. Nagios core. `http://nagios-br.com/nagios-core`. Accessed: 2016-11-28.
7. Douglas E Comer. *Redes de Computadores e Internet-6*. Bookman Editora, 2016.
8. Miguel Pupo Correia and Paulo Jorge Sousa. Segurança no software. *Lisboa: FCA*, 2010.
9. Kai-Oliver Detken, Thomas Rix, Carsten Kleiner, Bastian Hellmann, and Leonard Renners. Siem approach for a higher level of it security in enterprise networks. In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference on*, volume 1, pages 322–327. IEEE, 2015.
10. Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666, 2004.
11. Bhadreshsinh G Gohil, Rishi K Pathak, and Axaykumar A Patel. Federated network security administration framework. 2013.
12. Josune Hernantes, Gorka Gallardo, and Nicolas Serrano. It infrastructure-monitoring tools. *IEEE Software*, 32(4):88–93, 2015.
13. Yih Huang, Arun Sood, and Ravi K Bhaskar. Countering web defacing attacks with system self-cleansing. In *Proceedings of 7th Word Multiconference on Systemics, Cybernetics and Informatics*, pages 12–16, 2003.
14. Mohammed J Kabir. *Apache Server Bible*. IDG Books Worldwide, Inc., 1998.
15. R Kenig, D Manor, Z Gadot, and D Trauner. Ddos survival handbook, 2013.
16. John Levine. Dns blacklists and whitelists. Technical report, 2010.
17. Robert Love. Kernel korner: Intro to inotify. *Linux Journal*, 2005(139):8, 2005.
18. Miroslav Matýsek, Milan Adámek, M Kubalcík, and Miroslav Mihok. Monitoring of computer networks and applications using nagios. *Advances in Data Networks, Communications, Computers and Materials Monitoring*, pages 63–67, 2012.
19. Sophon Mongkolluksamee, Panita Pongpaibool, and Chavee Issariyapat. Strengths and limitations of nagios as a network monitoring solution. In *Proceedings of the 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010)*, volume 7, 2009.
20. Sérgio Manuel Maia Torres Moreira. *Monitorização de Redes e Sistemas Informáticos*. PhD thesis, Faculdade de Ciências da Universidade do Porto, 2014.
21. MA Pervilä. Using nagios to monitor faults in a self-healing environment. In *Seminar on Self-Healing Systems, University of Helsinki*, pages 1–6. Citeseer, 2007.
22. Hsiu-Lien Yeh, Yan-Fu Chen, Tsung-Tai Yeh, Pei-Chi Huang, Shin-Hao Liu, Hsin-Wen Wei, and Tsan-sheng Hsu. A monitoring system based on nagios for data grid environments. In *International Conference on Grid Computing and Applications*, 2011.
23. Lihua Yuan, Krishna Kant, Prasant Mohapatra, and Chen-Nee Chuah. Dox: A peer-to-peer antidote for dns cache poisoning attacks. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 5, pages 2345–2350. IEEE, 2006.