
TRABALHO PRÁTICO 1 *Tecnologias de Segurança*

TECHNICAL REPORT

📧 **Ivo Miguel Alves Ribeiro**
Mestrado em Engenharia Informática
Universidade do Minho
Pg53886
pg53886@alunos.uminho.pt

📧 **Henrique Fernades Ribeiro**
Mestrado Integrado em Engenharia Informática
Universidade do Minho
A95323
a95323@alunos.uminho.pt

10 de março de 2024

ABSTRACT

Este trabalho prático aborda duas fases distintas relacionadas à avaliação da segurança em sistemas e infraestruturas.

Na primeira parte, focamos na aplicação de técnicas de coleta passiva de informações como uma ferramenta analítica para avaliar a postura de segurança em ambientes reais. A segunda parte envolve a configuração de um ambiente de testes, no qual são utilizadas técnicas e ferramentas de varredura ativa são utilizadas como estratégia para identificar vulnerabilidades e fraquezas em um sistema remoto.

Ambas as partes contribuem para a fase de *footprinting* no contexto da atividade de testes de penetração.

Keywords segurança em sistemas e infraestruturas · coleta passiva · vulnerabilidades e fraquezas

1 Introdução

O presente relatório propõe uma abordagem abrangente na avaliação da segurança de sistemas e infraestruturas, dividindo-se em duas partes distintas, ambas inseridas na fase crucial de *footprinting* no âmbito da atividade de testes de penetração. A primeira parte concentra-se na aplicação de técnicas de coleta passiva de informações como uma ferramenta analítica para examinar a postura de segurança em ambientes reais. A segunda parte destaca a configuração de um ambiente de testes, onde técnicas e ferramentas de varredura ativa são utilizadas estrategicamente para identificar vulnerabilidades e fraquezas em sistemas remotos. Este artigo apresentará uma visão aprofundada dessas duas vertentes, explorando seu papel essencial na obtenção de *insights* valiosos para aprimorar a *cibersegurança*.

2 Parte A

Escolha duas empresas com operação comercial suportada por serviços on-line (uma grande corporação e um negócio local) e utilize técnicas de busca passiva de informação que permitam identificar detalhes sobre os seus sistemas e infra-estrutura. Descreva as estratégias usadas, os resultados obtidos e as possíveis diferenças de postura adotadas pelos administradores dos domínios estudados. Forneça uma análise crítica sobre os riscos associados às práticas identificadas. Enriqueça a sua análise apontando estratégias destinadas a fortalecer a postura de segurança destes domínios, especificamente, como resposta às técnicas e ferramentas de busca passiva.

Decidimos fazer a análise detalhada sobre os sistemas e infraestruturas da grande Google e para negócio local com serviços on-line optamos por um *website* de um negócio de contabilidade e gestão local.

Para ambas as empresas escolhidas iremos utilizar técnicas de busca passiva idênticas e realizar a análise segundo abordagens genéricas, respeitando os limites éticos e legais.

Começamos por utilizar mecanismos de pesquisa como *Shodan*, *Censys*, entre outros para obter informações sobre servidores, *IPs* e dispositivos associados ao domínio *google.com* e *jmalves.pt*.

Por fim, faremos uma análise passiva com recurso de dados disponíveis publicamente em *fóruns*, *blogs* e redes sociais que podem fornecer *insights* sobre a infraestrutura ou sistemas específicos.

- **Plataformas de busca passiva**

As plataformas de busca como *Shodan* e *Censys* são especializadas em coletar e indexar informações sobre dispositivos conectados à *internet*. Ao procurar informações sobre o domínio "*google.com*" e "*jmalves.pt*", podemos obter informações variadas, dependendo da configuração e exposição dos recursos online da empresa. Assim, exemplos de informações que temos acesso são:

- *Endereços IP* - É possível identificar os endereços *IP* associados ao domínio, assim como informações sobre servidores e serviços disponíveis nesses *IPs*.
- *Servidores web* - É possível descobrir quais os servidores web associados ao domínio, disponibilizando por vezes informações sobre versões de software, tecnologias web e módulos específicos.
- *Portas abertas* - É possível identificar quais são as portas abertas nos *IPs* associados, que indicam na sua maioria os serviços em execução em cada uma delas.
- *Protocolos utilizados* - É possível identificar os protocolos de rede.
- *Certificados SSL/TLS* - Informações sobre certificados *SSL/TLS* utilizados nos servidores são também disponibilizados, incluindo detalhes sobre data de expiração, algoritmos criptográficos, entre outros.
- *Dispositivos IoT e Conectados* - Acesso a uma lista de dispositivos conectados à *internet* associados ao domínio.
- *Possíveis Vulnerabilidades* - É possível descobrir versões de software desatualizadas que podem indicar possíveis vulnerabilidades. Com *Shodan*, em particular, é possível destacar sistemas com configurações de segurança fracas.
- *Histórico de Alterações* - Estes mecanismos podem ainda fornecer um histórico de alterações, permitindo rastrear mudanças na infraestrutura ao longo do tempo.

Como esperado o resultado destas pesquisas mostra um volume de dados muito maior referente à infraestrutura da *google* em comparação ao negócio local. Contudo conseguimos notar que para ambos os casos conseguimos obter informações relativas a endereços *IP* e sua localização geográfica, servidores, portas e protocolos utilizados com a utilização do *censys*, como descrito nas *figuras 6 e 7*.

Quanto à utilização do *shodan*, só conseguimos obter resultados relativos ao *google*, mas com ele também foi possível averiguar informações relativas a versões de *certificados SSL*, assim como datas de últimas atualizações, o que pode ser usado como vantagem para um ataque, como descrito na *figura 8*.

Para uma melhor análise, tentamos procurar em análises disponíveis em *fóruns*, *blogs* e redes sociais. Da parte da grande corporação (*Google*) já esperávamos encontrar uma maior quantidade de dados nestas ferramentas de comunicação entre a comunidade, mas, ainda assim, achamos bastante pertinente, daí o destaque deste caso particular da documentação detalhada da infraestrutura do *google cloud*, em que realçamos o facto de a própria corporação o descrever publicamente:

- Visão geral de segurança da infraestrutura
- Criptografia
- Como é realizada a autenticação

Vale mencionar que achamos muito particular e interessante esta transparência, talvez por nunca nos ter suscitado curiosidade de procurar sobre este assunto, contudo faz todo o sentido empresas como o Google adotarem uma abordagem transparente e responsável em relação à divulgação de questões de segurança na infraestrutura. Na verdade, acaba por demonstrar responsabilidade e comprometimento com a segurança, tanto para os clientes, quanto para a comunidade em geral. Por outro lado, ao compartilhar informações sobre problemas de segurança, permite que outras organizações evitem erros semelhantes e que possam aprimorar as suas próprias práticas de segurança, podendo ainda ajudar a minimizar os impactos negativos à reputação. Após nos debruçarmos um pouco mais sobre este tema através de mais algumas pesquisas, identificamos também pontos como o facto de a conformidade com normas e regulamentações poder exigir a divulgação de problemas de segurança, principalmente em setores altamente regulamentados, como o financeiro e da saúde. Para além disso, as empresas líderes muitas vezes consideram como parte de sua responsabilidade a contribuição para a comunidade de segurança, então disponibilizam vulnerabilidades, contribuem em projetos de código aberto ou participam em iniciativas relacionadas a segurança.

Quanto ao negócio local, utilizamos um método diferente de análise, porém bastante proativo, que foi o de perguntar presencialmente à organização sobre a infraestrutura, na qual destacamos um servidor físico com algum poder de processamento, mas principalmente focado em armazenamento de dados, e um serviço contratado a terceiros de administração desses recursos. Apesar de, aparentemente, se revelar numa estratégia bem implementada, após dois meses da primeira implementação deste sistema, foi identificado um problema de segurança que comprometia o poder de processamento. Problema este que se tratava de uma falha na área de cliente que até aí era disponibilizada por este serviço, onde um atacante conseguiu aproveitar uma falha para beneficiar do poder de processamento do servidor a seu favor e utilizá-lo para minerar criptomoedas. Esta anomalia foi identificada pelo facto de o sistema estar muito lento, até mesmo quando poucas pessoas acediam a esses recursos, o que alarmou a organização.

• Postura Adotada pelos Administradores:

A Google, sendo uma grande corporação, provavelmente mantém uma postura de segurança rígida. Faz uso de *firewalls*, *sistemas de deteção de intrusão*, e *práticas de segurança de rede avançadas*, com grandes recursos para apostar numa postura robusta no que toca a segurança. Assim como mencionamos acima aposta ainda numa postura transparente para com a comunidade, visando uma evolução e prevenção constante.

Já quanto ao negócio local, como os recursos são bem mais limitados, a segurança pode não ser uma prioridade principal. Vemos que, por causa disso, há dependência de soluções de segurança padrão oferecidas pelos provedores de hospedagem. Daí o facto, e em bom modo Português, "*casa roubada, trancas à porta*", que foi o que aconteceu no caso enunciado acima. O sistema foi atacado e os administradores tomaram partido de se certificar que traziam melhorias para as questões de segurança do serviço. Dois dias sem qualquer tipo de serviço on-line, que como é lógico, é um período de tempo mais que compreensível para resolver este tipo de problemas, a organização garante ainda que não foram afetadas nem divulgadas informações dos clientes, uma das principais preocupações da mesma.

- **Análise Crítica:**

O Google investe fortemente em segurança, mas a constante vigilância é crucial, uma vez que, por possuir tamanha importância global, torna-se um *"isco"* valioso não só para ciberataques de alto perfil, mas também para tentativas constantes de explorar suas vastas infraestruturas e dados sensíveis.

Quanto ao negócio local os riscos podem ser mais elevados devido à falta de recursos dedicados à segurança, logo a falta de atualizações regulares e uma monitorização constante pode deixar a empresa mais vulnerável, e causar sérios danos ao funcionamento do serviço.

- **Estratégias de Fortalecimento da Segurança:**

A *Google* tem como estratégias de fortalecimento da segurança a implementação de atualizações regulares de segurança, monitorização constante de fontes externas de informações, assim como estar envolvido em programas de recompensa por bugs para incentivar a comunidade a relatar vulnerabilidades. Uma prática muito bem estruturada e coesa que permite a longevidade dos seus serviços mesmo a uma larga escala.

Já no que toca ao negócio local, sabemos que a estratégia adotada pela organização foi a de desabilitar a área de cliente do serviço on-line e a informação que temos é que, até à data, procuram uma estratégia diferente com os serviços contratados para voltar a colocar esse serviço no ar. Para além dessa solução já estabelecida, recomendamos também o investimento em programas de terceiros de fonte segura, bem como a implementação de práticas de segurança padrão, atualizações regulares de software e instrução dos funcionários sobre práticas seguras, dentro do ambiente partilhado.

3 Parte B

Q1: Selecione um conjunto de ferramentas e técnicas de varredura ativa para identificar e detalhar vulnerabilidades e fraquezas para as quais o Sistema *Metasploitable* está exposto. A sua resposta deverá listar os serviços a correr neste sistema e as vulnerabilidades e/ou fraquezas relacionados a cada um. Para os serviços com diferentes vulnerabilidades, escolha a mais recente ou a mais grave. Importante: Para esta questão, não será permitido o uso de Scanners de Vulnerabilidades (por exemplo, *OpenVAS* ou *Nessus*). Uma lista abrangente de ferramentas pode ser consultada em www.sectools.org

O uso do *Nmap* no *Metasploitable* pode fornecer informações valiosas sobre os serviços em execução e portas abertas como por exemplo *SSH*, *Apache HTTP Server*, *MySQL*, entre outros. Para além disso, é possível identificar versões desses serviços, permitindo uma análise mais detalhada que é útil para determinar se há vulnerabilidades conhecidas associadas a versões específicas.

Nmap pode usar *scripts* de varredura de vulnerabilidades (*NSE scripts*) para detetar ativamente algumas vulnerabilidades já conhecidas no sistema e ainda pode ajudar a identificar *hosts* ativos na rede, mesmo que estejam mascarados por *firewalls*, assim como, através de uma análise de portas e serviços, é possível identificar dispositivos *IoT* ou outros dispositivos conectados à rede.

```
(kali@kali)~$ nmap -sV 172.24.14.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-05 10:20 EST
Nmap scan report for 172.24.14.2
Host is up (0.00095s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3000/tcp   open  http             WEBrick httpd 1.3.1 (Ruby 2.3.3 (2016-11-21))
3306/tcp   open  mysql            MySQL 5.5.20-log
3389/tcp   open  ssl/ms-wbt-server?
4848/tcp   open  ssl/http         Oracle Glassfish Application Server
7676/tcp   open  java-message-service
8009/tcp   open  ajp13            Apache Jserv (Protocol v1.3)
8022/tcp   open  http             Apache Tomcat/Coyote JSP engine 1.1
8031/tcp   open  ssl/unknown
8080/tcp   open  http             Sun GlassFish Open Source Edition 4.0
8181/tcp   open  ssl/intermapper?
8383/tcp   open  http             Apache httpd
8443/tcp   open  ssl/https-alt?
8080/tcp   open  http-proxy
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|   State: LIKELY VULNERABLE
|   IDS: CVE:CVE-2007-6750
|   Slowloris tries to keep many connections to the target web server open and hold
|   them open as long as possible. It accomplishes this by opening connections to
|   the target web server and sending a partial request. By doing so, it starves
|   the http server's resources causing Denial Of Service.
|
|   Disclosure date: 2009-09-17
|   References:
|     http://ha.ckers.org/slowloris/
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|   http-open-proxy: Proxy might be redirecting requests
```

Figura 1: Identificação de vulnerabilidades com Nmap

Optamos por descrever algumas das vulnerabilidades detetadas ativamente segundo esta ferramenta.

Serviço -> *open-ssh*

Porta -> 22

Versão -> *OpenSSH 7.1* (protocol 2.0)

Vulnerabilidades (CVE) -> CVE-2023-51385

Impacto -> *Medium*

Descrição -> Esta vulnerabilidade descreve uma injeção de comandos (*OS command injection*) que pode ocorrer quando um nome de usuário ou nome de *host* contiver caracteres especiais de shell. No contexto do *Git*, essa vulnerabilidade pode ser explorada em submódulos de repositórios *Git*, onde um nome de usuário ou nome de *host* malicioso, contendo metacaracteres de shell que pode ser utilizado para injetar comandos maliciosos.

Serviço -> Microsoft Windows RPC

Porta -> 135

Versão -> Microsoft Windows RPC)

Vulnerabilidades (CVE) -> CVE-2020-1113

Impacto -> High

Descrição -> A vulnerabilidade CVE-2020-1113 é uma falha de segurança no Microsoft Windows, relacionada ao serviço *Task Scheduler*. Essa vulnerabilidade permite a um invasor contornar a verificação adequada das conexões de cliente sobre o protocolo *RPC (Remote Procedure Call)*. Isso pode ser explorado para executar código malicioso no sistema comprometido, representando um risco crítico à segurança.

Serviço -> Microsoft Windows netbios

Porta -> 139

Versão -> Microsoft Windows netbios-ssn)

Vulnerabilidades (CVE) -> CVE-2017-0174

Impacto -> Medium

Descrição -> A CVE-2017-0174 é uma vulnerabilidade no protocolo *Windows NetBIOS*. Essa falha permite a um invasor causar uma negação de serviço (*DoS*) explorando uma falha no tratamento de pacotes *NetBIOS* pelo sistema operacional, resultando na interrupção dos serviços ou recursos do sistema alvo.

Serviço -> Microsoft Windows Server

Porta -> 445

Versão -> Microsoft Windows Server 2008 R2 - 2012 microsoft-ds)

Vulnerabilidades (CVE) -> CVE-2018-8553

Impacto -> High

Descrição -> A CVE-2018-8553 é uma vulnerabilidade que permite que um invasor execute código remotamente, explorando a maneira como os Componentes Gráficos da Microsoft lidam com objetos na memória do sistema. Como resultado, um invasor pode potencialmente assumir o controle do sistema comprometido.

Serviço -> WEBrick

Porta -> 3000

Versão -> WEBrick *httpd* 1.3.1 com versões *Ruby* anteriores a 2.2.10, 2.3.x antes de 2.3.7, 2.4.x antes de 2.4.4, 2.5.x antes de 2.5.1 e 2.6.0-preview1

Vulnerabilidades (CVE) -> CVE-2018-8777

Impacto -> High

Descrição -> Um atacante pode explorar essa vulnerabilidade enviando uma requisição *HTTP* com um *header* manipulado para o servidor *WEBrick* ou um corpo manipulado para o *servidor/handler WEBrick*, resultando em uma negação de serviço devido ao consumo excessivo de memória.

Serviço -> MySQL

Porta -> 3306

Versão -> MySQL 5.5.20-log

Vulnerabilidades (CVE) -> CVE-2012-0882

Impacto -> High

Descrição -> A CVE-2012-0882 é uma vulnerabilidade relacionada a um estouro de *buffer* no *yaSSL*, utilizado no *MySQL*. Isso permite que atacantes remotos executem código arbitrário através de vetores não especificados, conforme demonstrado pelo *VulnDisco Pack Professional 9.17*.

Serviço -> Oracle GlassFish Server

Porta -> 4848

Versão -> Oracle GlassFish Server 3.1.2.18 e versões anteriores

Vulnerabilidades (CVE) -> CVE-2021-3314

Impacto -> Medium

Descrição -> Esta vulnerabilidade permite ataques de *Cross-Site Scripting (XSS)* através do arquivo */common/logViewer/logViewer.jsf*. Um usuário malicioso pode persuadir um administrador a fornecer conteúdo perigoso para a página vulnerável, que é então refletido de volta ao usuário e executado pelo navegador web. O vetor de ataque mais comum envolve a inclusão de conteúdo malicioso como parâmetro em um URL divulgado publicamente ou enviado por e-mail diretamente às vítimas.

Serviço -> *Apache jserv/Tomcat*

Porta -> 8022

Versão -> *Apache Tomcat* nas versões 9.0.0.M1 até 9.0.0.30, 8.5.0 até 8.5.50 e 7.0.0 até 7.0.99

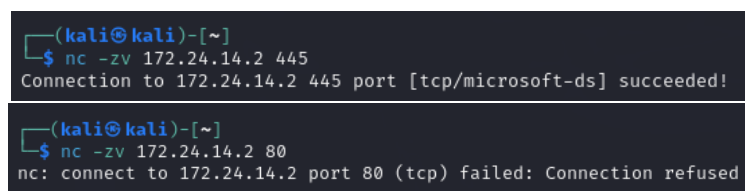
Vulnerabilidades (CVE) -> CVE-2020-1938

Impacto -> *Critical*

Descrição -> Esta vulnerabilidade permite a exploração de conexões *AJP*, que por padrão estavam habilitadas e ouvindo em todos os endereços IP configurados. Isso possibilitava a um atacante realizar a recuperação de arquivos arbitrários e processar qualquer arquivo na aplicação web como *JSP*, podendo levar à execução remota de código, logo é necessária a mitigação se a porta *AJP* estiver acessível a usuários não confiáveis.

Essencialmente, com *Netcat*, apesar de ser uma ferramenta versátil para interação e transferência de dados em redes, foi usado por nós apenas para testes de conectividade simples, onde verificamos a acessibilidade de um *host* e porta específicos. Testamos esta ferramenta para algumas das portas mais utilizadas como 22, 80, 119, 443, 445.

Das quais identificamos resultados como os disponibilizados nestas imagens.



```
(kali@kali)-[~]  
$ nc -zv 172.24.14.2 445  
Connection to 172.24.14.2 445 port [tcp/microsoft-ds] succeeded!  
  
(kali@kali)-[~]  
$ nc -zv 172.24.14.2 80  
nc: connect to 172.24.14.2 port 80 (tcp) failed: Connection refused
```

Figura 2: Identificação de vulnerabilidades com *Netcat*

A conexão foi bem-sucedida na porta 445, que é normalmente associada ao protocolo *SMB* (*Server Message Block*), usado para compartilhar de arquivos e recursos em redes Windows. Um "*Connection Succeeded*" indica que a porta está aberta e aceitando conexões no *host-alvo*. Isso pode ser um indicativo de que o serviço *SMB* está em execução e acessível.

Já na porta 80, a conexão foi recusada, o que indica que a porta está fechada no *host-alvo*, ou que não há nenhum serviço ou aplicação à espera de conexões na porta 80.

Q2: Discuta os resultados globais do processo de varredura ativa ao Sistema *Metasploitable*. Avalie também as diferenças entre o resultado do sistema automático de identificação de vulnerabilidades e o resultado que obteve no item Q1 da Parte B deste enunciado.

Analisando os resultados do processo de varredura ativa (*Nessus*) no Sistema *Metasploitable*, observamos que existem várias vulnerabilidades críticas, principalmente relacionadas com o ao servidor *web Apache* e com o software *ManageEngine Desktop Central*. Essas vulnerabilidades incluem várias falhas de segurança, como execução remota de código e escalonamento de privilégios.

A maioria das vulnerabilidades encontradas é classificada como crítica, com pontuações *CVSS* (*Common Vulnerability Scoring System*) altas, o que indica a gravidade das falhas de segurança. As vulnerabilidades abrangem várias versões do servidor Apache, sugerindo que o sistema *Metasploitable* possa estar a usar versões desatualizadas do software. Além das vulnerabilidades no Apache, também foram identificadas vulnerabilidades no *ManageEngine Desktop Central*, destacando a presença de múltiplos pontos de acesso com potencial para invasões. A alta severidade das vulnerabilidades e os riscos associados representam uma ameaça significativa à segurança do sistema, com potencial para exploração maliciosa, comprometimento dos dados e intrusão não autorizada.

Comparando com os resultados obtidos na pergunta Q1 da Parte B, é importante notar que os resultados do *Nessus* são muito mais abrangentes e detalhados, fornecendo uma visão mais completa das vulnerabilidades presentes no sistema. Enquanto as ferramentas que usamos na primeira pergunta, apesar de conseguirmos encontrar vulnerabilidades com alguma severidade, não conseguimos perceber a dimensão das falhas de segurança que o sistema *Metasploitable* contém.

Q3: Examine o output do IDS e escolha dois eventos identificados como tráfego anômalo. Para cada evento escolhido, identifique o respetivo tráfego capturado via Analisador de tráfego e o descreva. Se possível, inclua o CVE da vulnerabilidade e o método de identificação usado pelo scanner.

Como resultado do IDS escolhido (*suricata*) foram obtidos os seguintes alertas:

```
ET EXPLOIT [FIREEYE] Suspicious Pulse Secure HTTP Request (CVE-2021-22893)
M1 [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.24.14.1:42022 -> 172.24.14.2:8022
ET HUNTING Possible Apache log4j RCE Attempt - Any Protocol TCP (Outbound)
(CVE-2021-44228) [**] [Classification: Misc activity] [Priority: 3] {TCP} 172.24.14.1:58078 -> 172.24.14.2:8019
```

Figura 3: Tráfego anômalo identificado pelo *suricata*

Aprofundando os eventos destacados na **figura 3** através da ferramenta *wireshark* obtivemos os seguintes dados relatados na **figura 4**.

```
28172 556.482057224 172.24.14.1 172.24.14.2 TCP 66 42022 -> 8022 [ACK] Seq=298
Frame 28172: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
  Section number: 1
    Interface id: 0 (eth1)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 4, 2024 08:34:41.484685704 EST
    UTC Arrival Time: Mar 4, 2024 13:34:41.484685704 UTC
    Epoch Arrival Time: 1709559281.484685704
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000079479 seconds]
    [Time delta from previous displayed frame: 0.000079479 seconds]
    [Time since reference or first frame: 556.482057224 seconds]
    Frame Number: 28172
    Frame Length: 66 bytes (528 bits)
    Capture Length: 66 bytes (528 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
  Ethernet II, Src: PCSSystemtec_1b:dc:f4 (08:00:27:1b:dc:f4), Dst: PCSSystemtec_90:26:74 (08:00:27:90:26:74)
  Internet Protocol Version 4, Src: 172.24.14.1, Dst: 172.24.14.2
  Transmission Control Protocol, Src Port: 42022, Dst Port: 8022, Seq: 298, Ack: 1449, Len: 0
```

Figura 4: Pacotes no *wireshark* correspondentes ao primeiro evento escolhido

O primeiro evento corresponde a vulnerabilidade de *bypass* de autenticação que permite a execução remota de código por um usuário não autenticado através do recurso do *Windows File Share Browser e Pulse Secure Collaboration*.

```
11830 45.453722264 172.24.14.1 172.24.14.2 TCP 66 58078 -> 8019 [RST, ACK] Seq=298
Frame 11830: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth1, id 0
  Ethernet II, Src: PCSSystemtec_1b:dc:f4 (08:00:27:1b:dc:f4), Dst: PCSSystemtec_90:26:74 (08:00:27:90:26:74)
  Internet Protocol Version 4, Src: 172.24.14.1, Dst: 172.24.14.2
  Transmission Control Protocol, Src Port: 58078, Dst Port: 8019, Seq: 86, Ack: 2, Len: 0
    Source Port: 58078
    Destination Port: 8019
    [Stream index: 5071]
    [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 0]
    Sequence Number: 86 (relative sequence number)
    Sequence Number (raw): 2731208655
    [Next Sequence Number: 86 (relative sequence number)]
    Acknowledgment Number: 2 (relative ack number)
    Acknowledgment number (raw): 484825666
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x014 (RST, ACK)
    0000 .... = Reserved: Not set
    ....0... = Accurate ECN: Not set
    ....0... = Congestion Window Reduced: Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0... = Reset: Set
    ....0... = Syn: Not set
    ....0... = Fin: Not set
    [TCP Flags: .....A.R..]
    Window: 251
    [Calculated window size: 32128]
    [Window size scaling factor: 128]
    Checksum: 0x745a [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
```

Figura 5: Pacotes no *wireshark* correspondentes ao segundo evento escolhido

Enquanto o segundo evento corresponde a uma vulnerabilidade no *Apache Log4j2*, que nas versões mais antigas apresenta uma falha de segurança relacionada ao uso de recursos *JNDI* na configuração, mensagens de *log* e parâmetros, permitindo a execução de código arbitrário por um atacante que controle as mensagens de *log* ou seus parâmetros, especialmente quando a substituição de mensagens está ativada.

Q4: Observe que algumas notificações do IDS não possuem vulnerabilidade correspondente no relatório do Scanner de vulnerabilidades. Apresente e discuta as possíveis razões para estas diferenças.

As discrepâncias entre as notificações do *IDS* e as vulnerabilidades relatadas pelo Scanner de Vulnerabilidades ocorrerem por várias razões. Em primeiro lugar, esses sistemas usam métodos diferentes para detetar ameaças. Enquanto o *IDS* se baseia em padrões de tráfego, assinaturas ou comportamentos suspeitos, o Scanner de Vulnerabilidades examina especificamente o sistema em busca de falhas conhecidas de software e configurações inseguras. Portanto, o *IDS* pode identificar atividades que não correspondem diretamente a uma vulnerabilidade específica detetada pelo Scanner de Vulnerabilidades. Além disso, o *IDS* pode gerar falsos positivos ou falsos negativos, e da mesma forma, o Scanner de Vulnerabilidades pode falhar em detetar certas vulnerabilidades ou gerar alertas falsos. As diferenças entre os resultados podem ser atribuídas a esses erros de deteção.

Q5: Escolha três vulnerabilidades identificadas pelo Scanner de vulnerabilidades, sendo, pelo menos, uma classificada como High/Critical e uma classificada como Medium. Pesquise a documentação referente às formas de corrigir a fonte do problema e efetue os procedimentos necessários para tal. Ao final dos procedimentos escolhidos para cada vulnerabilidade, execute uma nova varredura para garantir que estas já não são identificadas. Discuta a solução dada e inclua os ficheiros resultantes da varredura antes e depois das respetivas correções.

Para este exercício escolhemos duas vulnerabilidades classificadas como *Medium* e uma *High/Critical*.

- Solução para a vulnerabilidade "*SMB Signing not required*"

A solução para esta vulnerabilidade é bastante simples de implementar, sendo apenas necessário alterar a configuração de "*Always Digitally Sign Communication*", conforme indicado na **figura 8**. Ativar esta opção é crucial para fortalecer a segurança, garantindo que todas as comunicações sejam digitalmente assinadas, mitigando assim a vulnerabilidade identificada, ação descrita pela **figura 9**.

- Solução para a vulnerabilidade "*SSH Terrapin Prefix Truncation Weakness*"

A segunda vulnerabilidade selecionada, *CVE-2023-48795*, representa uma séria ameaça à segurança, viabilizando a execução remota de código. Esta falha permite que invasores realizem operações maliciosas à distância, o que elevava a importância de um combate urgente para corrigir esta vulnerabilidade de maneira a proteger a integridade e a segurança dos sistemas afetados. Uma solução para mitigar esta vulnerabilidade fornecida pelo *Nessus*, como representado na **figura 10**, foi a de desabilitar os algoritmos afetados, sendo esta a solução que adotamos, como descrevemos na **figura 11**.

- Solução para a vulnerabilidade "*Apache*"

Para a vulnerabilidade classificada como *High/Critical*, tivemos problemas em escolher uma, pois todas as vulnerabilidades com essa classificação tinham como solução atualizar um dado serviço. Porém, e como apenas é suposto que a máquina tenha acesso à rede interna, não é possível realizar algumas das atualizações, então optamos por desabilitar um serviço como substituição à atualização. Como descrito na **figura 12**, a solução para mitigar a vulnerabilidade é a de atualizar este serviço, porém, e como acima referido, ao invés da atualização do serviço decidimos desabilitá-lo, como descrito na **figura 13**.

Depois destas alterações, procedemos a uma nova varredura do Sistema *Metasploitable*. Esta varredura apresenta menos vulnerabilidades e nas quais não estão presentes nenhuma das referidas acima, como descrito na comparação presente na **figura 14**. Com isto, concluímos que as soluções que aplicamos resolveram as suas vulnerabilidades.

Referências

Parte A

- <https://search.censys.io/>
- <https://www.shodan.io/>
- <https://cloud.google.com/docs/security/infrastructure/design?hl=pt-br>

Parte B

- <https://sectools.org/>
- <https://nvd.nist.gov/>

Anexos

Parte A

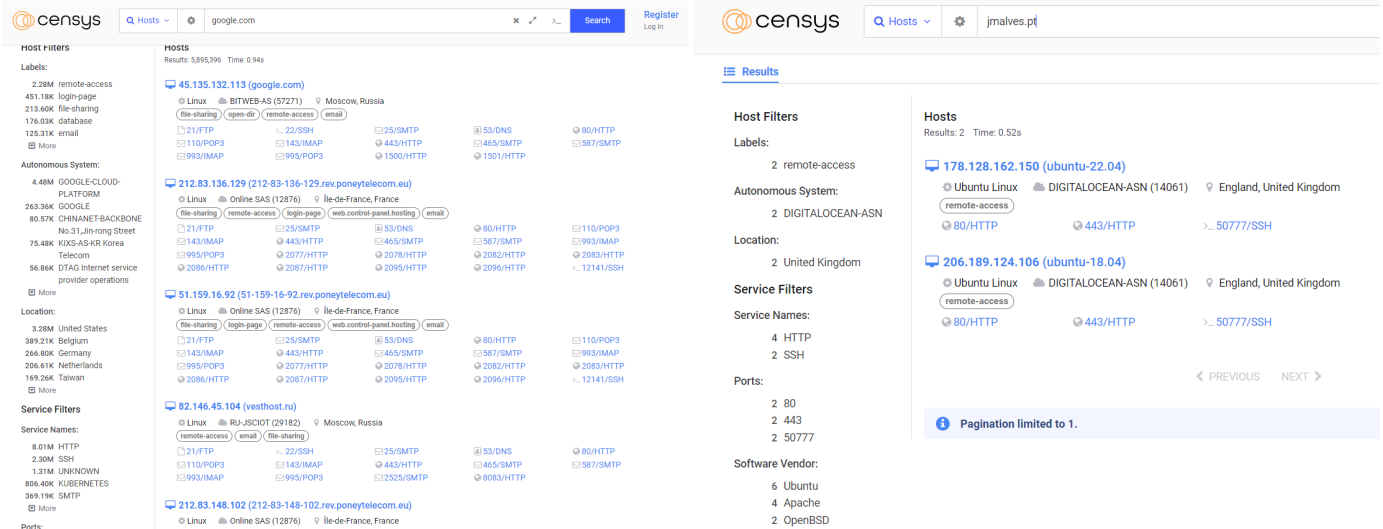


Figura 6: Amostra da pesquisa no *Censys* sobre informações do Google e do Negócio local

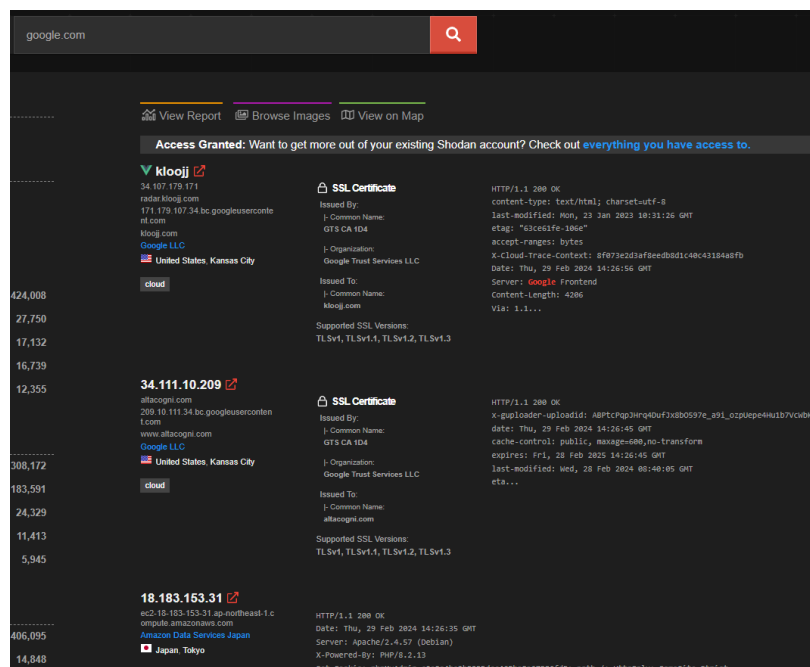


Figura 7: Amostra da pesquisa no *Shodan* sobre informações do Google

Parte B

MEDIUM

SMB Signing not required

Description

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

Solution

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

Figura 8: Descrição da vulnerabilidade do *SMB*

Figura 9: Solução da vulnerabilidade do *SMB*

MEDIUM

SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)

Description

The remote SSH server is vulnerable to a man-in-the-middle prefix truncation weakness known as Terrapin. This can allow a remote, man-in-the-middle attacker to bypass integrity checks and downgrade the connection's security.

Note that this plugin only checks for remote SSH servers that support either ChaCha20-Poly1305 or CBC with Encrypt-then-MAC and do not support the strict key exchange countermeasures. It does not check for vulnerable software versions.

Solution

Contact the vendor for an update with the strict key exchange countermeasures or disable the affected algorithms.

Figura 10: Descrição da vulnerabilidade do *SSH*

The screenshot shows the Windows Services console with the 'Services (Local)' window open. The 'OpenSSH Server' service is selected in the left-hand list. The right-hand pane displays the details for this service in a table:

Name	Description	Status	Startup Type	Log On As
Network List Service	Identifies t...	Started	Manual	Local Service
Network Location A...	Collects an...	Started	Automatic	Network S...
Network Store Inte...	This servic...	Started	Automatic	Local Service
OpenSSH Server			Automatic	.\sshd_se...
Performance Count...	Enables re...		Manual	Local Service
Performance Log...			Manual	Local Service

Figura 11: Solução da vulnerabilidade do *SSH*

CRITICAL Apache 2.2.x < 2.2.33-dev / 2.4.x < 2.4.26 Multiple Vulnerabilities

Description

According to its banner, the version of Apache running on the remote host is 2.2.x prior to 2.2.33-dev or 2.4.x prior to 2.4.26. It is, therefore, affected by the following vulnerabilities :

- An authentication bypass vulnerability exists due to third-party modules using the `ap_get_basic_auth_pw()` function outside of the authentication phase. An unauthenticated, remote attacker can exploit this to bypass authentication requirements. (CVE-2017-3167)
- A NULL pointer dereference flaw exists due to third-party module calls to the `mod_ssl ap_hook_process_connection()` function during an HTTP request to an HTTPS port. An unauthenticated, remote attacker can exploit this to cause a denial of service condition. (CVE-2017-3169)
- A NULL pointer dereference flaw exists in `mod_http2` that is triggered when handling a specially crafted HTTP/2 request. An unauthenticated, remote attacker can exploit this to cause a denial of service condition. Note that this vulnerability does not affect 2.2.x. (CVE-2017-7659)
- An out-of-bounds read error exists in the `ap_find_token()` function due to improper handling of header sequences. An unauthenticated, remote attacker can exploit this, via a specially crafted header sequence, to cause a denial of service condition. (CVE-2017-7668)
- An out-of-bounds read error exists in `mod_mime` due to improper handling of Content-Type response headers. An unauthenticated, remote attacker can exploit this, via a specially crafted Content-Type response header, to cause a denial of service condition or the disclosure of sensitive information. (CVE-2017-7679)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

Solution

Upgrade to Apache version 2.2.33-dev / 2.4.26 or later.

Figura 12: Descrição das vulnerabilidades do Apache

Services (Local)

Apache Tomcat 8.0 Tomcat8

Description:
Apache Tomcat 8.0.33 Server -
<http://tomcat.apache.org/>

Name	Description	Status	Startup Type	Log On As
Apache Tomcat 8.0...	Apache To...	Disabled	Manual	Local System
Application Experie...	Processes ...	Started	Manual	Local System
Application Identity	Determines...	Manual	Manual	Local Service
Application Informa...	Facilitates ...	Manual	Manual	Local System

Figura 13: Solução das vulnerabilidades do Apache

Scan Summary Hosts 1 Vulnerabilities 46 Remediations 4 Notes 3 History 2

Filter Search Vulnerabilities 46 Vulnerabilities

Sev	CVSS	VPR	Name
MIXED	Apache HTTP Server (Multiple Issues)
MIXED	Zohocorp Manageengine Desktop Central (Multiple Issues)
MIXED	Apache Httpd (Multiple Issues)
MIXED	Microsoft Windows (Multiple Issues)
MEDIUM	6.5	2.5	Remote Desktop Protocol Server Man-in-the-Middle Weakness
MEDIUM	4.3 *	...	Web Application Potentially Vulnerable to Clickjacking
MIXED	Microsoft Windows (Multiple Issues)
MIXED	Openbsd Openssh (Multiple Issues)
MIXED	SMB (Multiple Issues)
LOW	2.6 *	...	Terminal Services Encryption Level is not FIPS-140 Compliant
MIXED	Web Server (Multiple Issues)

Scan Summary Hosts 1 Vulnerabilities 35 Remediations 1 Notes 29 History 2

Filter Search Vulnerabilities 35 Vulnerabilities

Sev	CVSS	VPR	Name
MIXED	Microsoft Windows (Multiple Issues)
MIXED	SSL (Multiple Issues)
MIXED	IETF Md5 (Multiple Issues)
MEDIUM	6.5	...	Remote Desktop Protocol Server Man-in-the-Middle Weakness
MEDIUM	6.5	...	TLS Version 1.0 Protocol Detection
MIXED	Microsoft Windows (Multiple Issues)
LOW	2.6 *	...	Terminal Services Encryption Level is not FIPS-140 Compliant
INFO	SMB (Multiple Issues)
...	TLS (Multiple Issues)

Figura 14: Resultados da varredura antes(a esquerda) e depois(a direita) das soluções encontradas