Vulnerability Management

Penetration Tests

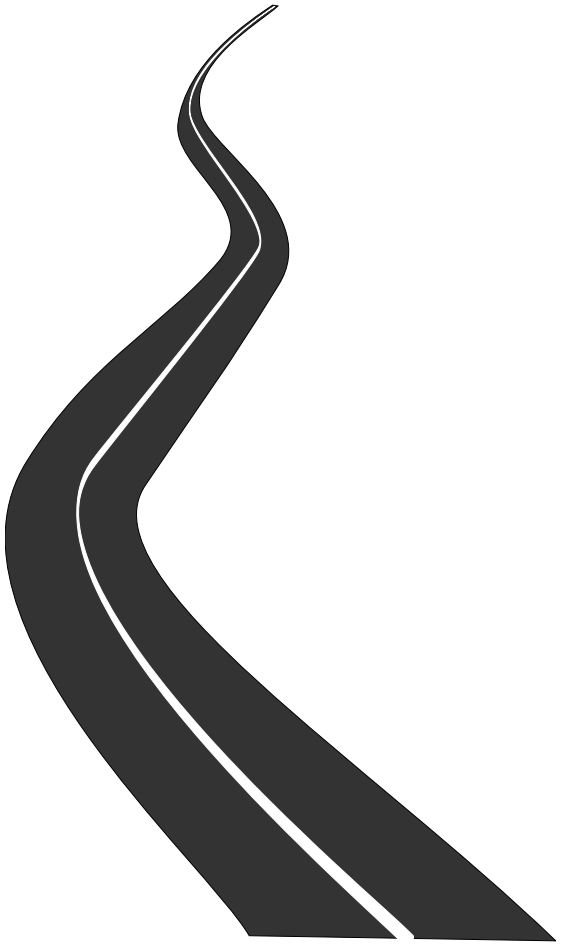
Klaus Möller
WP8-T1

Webinar, 13th of September 2021

Public

www.geant.org

The Road Ahead: What we will cover today



- Security Testing & Penetration Tests
- Cyber Kill Chain
- Penetration Test Execution Standard (PTES)
  - Pre-Engagement
  - Information Gathering
  - Threat Modeling
  - Vulnerability Analysis
  - Exploitation
  - Post-Exploitation
  - Report
- Penetration Tests vs. Vulnerability Assessments
- Sample Penetration Test Tools

DFN DEUTSCHES FORSCHUNGSNETZ DFN CERT®

GÉANT

Security Testing

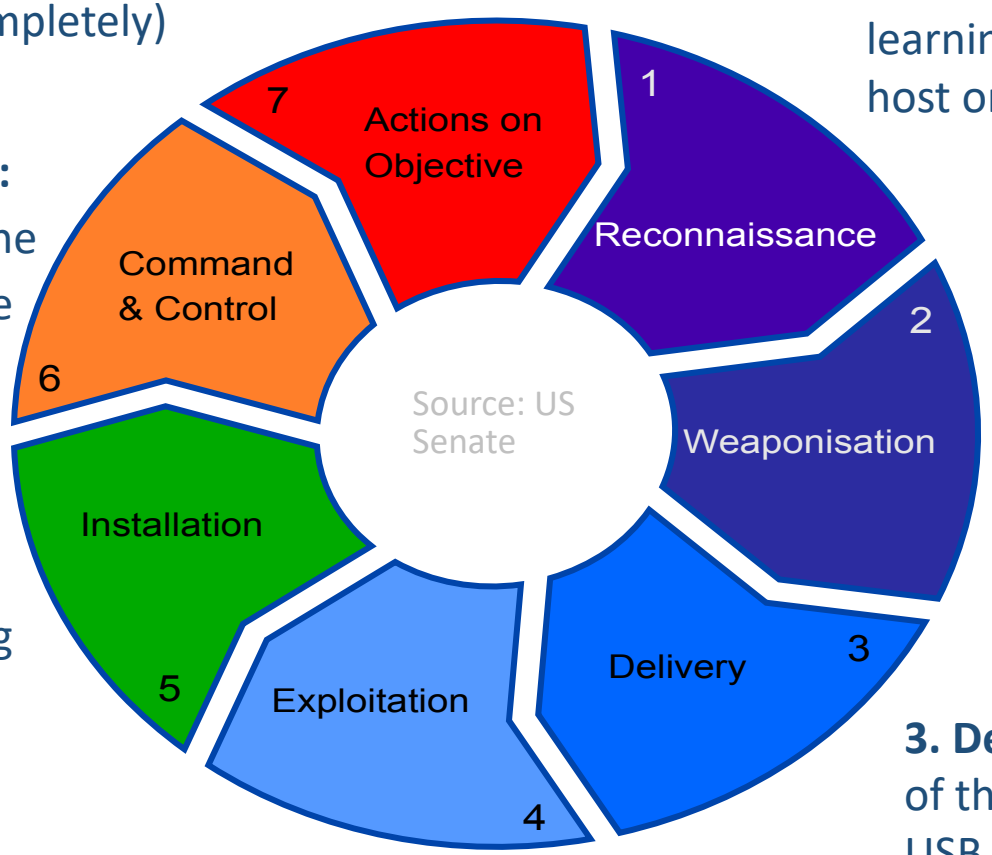
- *Software security testing is the process of assessing and testing a system to discover security risks and vulnerabilities of the system and its data* (OWASP)
- **Vulnerability Assessment**
  - The system is scanned and analyzed for security issues
- **Code Review**
  - The system code undergoes a detailed review and analysis looking specifically for security vulnerabilities
- **Runtime Testing**
  - The system undergoes analysis and security testing from an end-user
- **Penetration Testing**
  - The system undergoes analysis and attack from simulated malicious attackers

Attackers Workflow: (Intrusion|Cyber) Kill Chain

7. Actions or Objections: The attacker meets his/her goal (e.g. stealing information, gaining elevated privileges or damaging the host completely)

1. Reconnaissance: Collecting information and learning about the internal structure of the host organization

6. Command & Control: Setting up controls so the attacker can have future access to the host's network



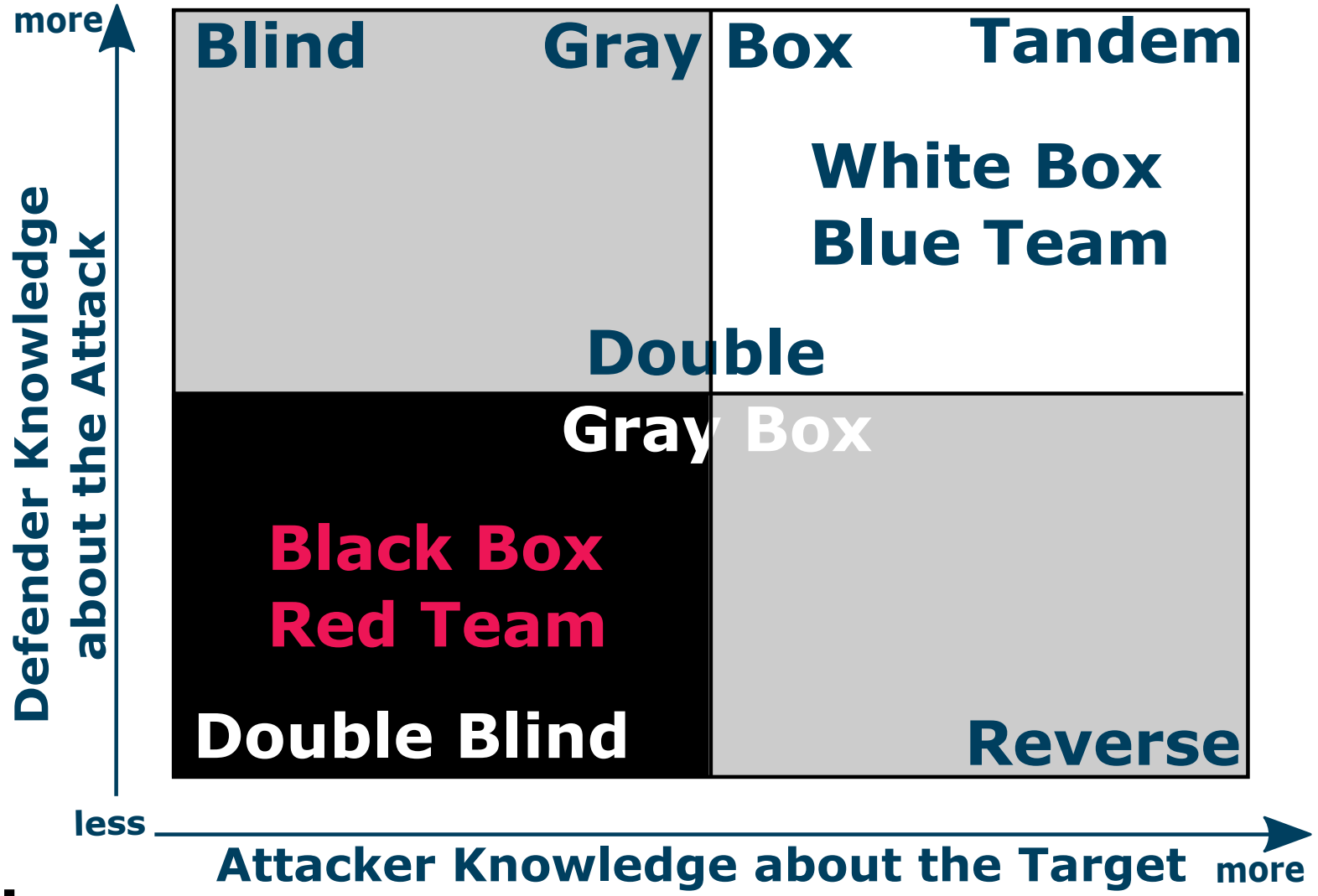
2. Weaponization: How the attacker packages the threat for delivery

5. Installation: Installing the actual malware

3. Delivery: The actual delivery of the threat (via email, web, USB, etc.)

4. Exploitation: Once the host is compromised, the attacker can take advantage and conduct further attacks

Types of Penetration Tests



Penetration Test Execution: Pre-Engagement

- Meeting with the customer
  - Get to know each other
  - Understand the situation
- Define the scope of the penetration test
  - I. e. what should be tested and what not
  - IP-Address(es| ranges), domains, applications, URLs, …
  - Knowledge of the penetration tester (attacker) about the target?
  - Knowledge of the defender (admins) about the attacks/attackers?
- Logistics
  - Rooms, Network connection, Keys, ...
  - Emergency Contacts, Encryption, ...
  - Time frame

Attacker Knowledge about the Target

- Black Box: Attacker has (almost) no knowledge about the system or application to test
  - Except whats needed to start the test (IP-Address, URL, …)
  - and where to stop (Scope)
- Gray Box: Attacker has some knowledge about the test object
  - Information about the architecture, what the system is used for, etc.
  - What kind of data is processed on the system
  - (some) Documentation, ...
- White Box: Attacker has extensive knowledge about the system or application being tested
  - May include the source code

Defender Knowledge about the Attack

- No knowledge (Blind):
  - No information when the attack will take place or what will be attacked
  - Sometimes vague information **that** a pentest will take place in a certain time window
- Some knowledge: Some details will be known by the defenders
  - IP-Addresses from where the attackers will come
  - Target of the attack, etc.
- Complete knowledge: Defenders know precisely where, when and how the attacker will strike
- Red/Blue team: attacking/defending side (by convention)

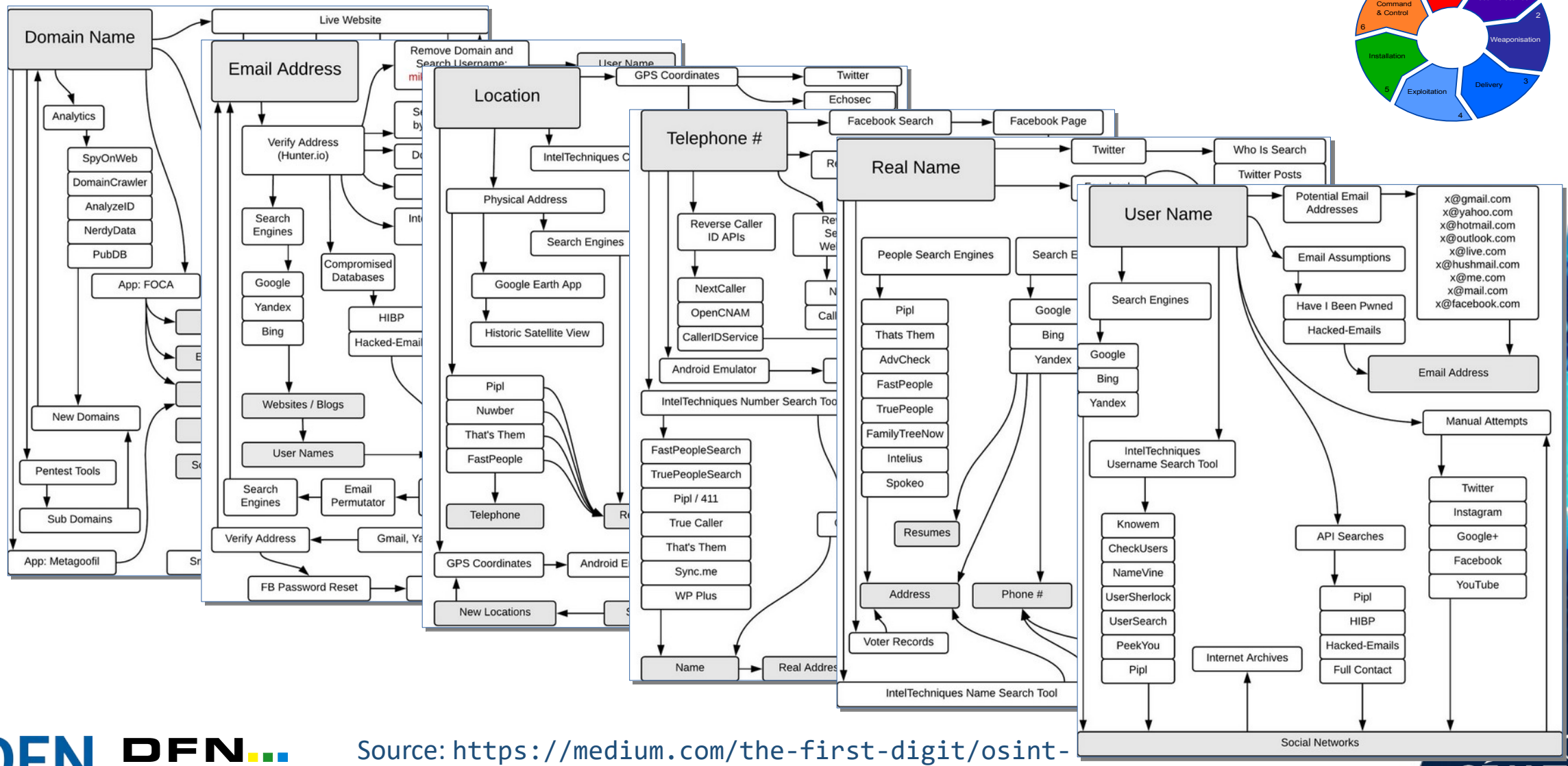
Legal Advice

- **The activities of a penetration test are usually illegal!**
  - I. e. breaking into systems or applications - details depend on your legislation
  - Other laws may also apply: burglary, wiretapping, privacy protection, etc.
  - Penetration tests can be legal, when conducted properly!
- We are not lawyers - check with yours before doing anything
  - Get **written** mandate/contract before you start
  - From **all** parties involved (Cloud providers, Security providers, etc.)
- Check insurance coverage (tester and maybe customer)
  - In case something is accidentally broken
  - Limit scope to minimize potential damage

Penetration Test Execution: Intelligence Gathering

- Step 1 of the Kill Chain - get information about the target
- Information from the client (gray-/white box tests)
  - Network plans, Communication matrices
  - General information about the test subject,
  - Security measures already in place
- Open source intelligence
  - Search engines (Shodan, Google, etc.)
  - DNS, WHOIS, BGP, Website, etc.
  - Social networks
  - Public databases (Court records, etc.)
- Physical presence
  - Site visits, dumpster diving, etc.

OSINT Sample Workflows



Source: https://medium.com/the-first-digit/osint-how-to-find-information-on-anyone-5029a3c7fd56

Penetration Test Execution: Intelligence Gathering (cont.)

- Footprinting
  - Portscans
  - Wireless networks (WiFi, IoT, etc.)
  - DNS sweeps, zone-transfers, brute-forcing
  - Banner grabbing, SMTP bounces, SNMP, ...
- Identification of security mechanisms
  - Network: Firewalls, IDS, Load-balancing, Traffic shaping , etc.
  - Host: Stack/Heap protection, RBAC, AV-Scanners, Kernel hardening, App whitelists, etc.
  - Find ways to bypass these
  - Finding the weakest link

Penetration Test Execution: Threat Modeling

- Identify and document
  - Assets - what needs to be protected
    - Business assets: employee/student records, research plans/results, financial data, …
    - Business processes: Infrastructure (PCs, server, networks), assets supporting processes
  - Attackers
    - Threat communities - i. e. types of attackers
      - Internal: Employees, Management, Admins, End users, ...
      - External: Nation states, script kiddies, organized crime, competitors, …
    - Threat capabilities - what the communities are capable of doing
    - Motivation: Money, political, …
- Goal: Refine understanding of both sides what the test is about

Penetration Test Execution: Vulnerability Analysis

- Still at step 1 of the Kill Chain
- Analyze detected/identified services and processes for vulnerabilities
- Decide which avenues for attack to take
  - Choose path of least resistance
  - Humans are often the weakest link → Social Engineering
  - Align with threat model
  - Check if within scope
- Technical
  - Banner grabbing, document metadata etc.
  - Vulnerability scanners, e. g. OpenVAS, Nessus, etc.
  - Application scanners/fuzzers, e. g. ZAP, skipfish, sqlmap, etc.
  - Research in vulnerability databases, e. g. exploitdb.com, etc.

Penetration Test Execution: Exploitation

- Step 2 - 4 of the Kill Chain
- Create a (proof of concept) exploit and use it
- Take security measures into account
    - Encode/pack exploit code for transmission
    - Evade IDS, AV-Scanner, etc.
- Use as proof the vulnerability really exists
    - May not be necessary, depending on agreement with customer

Penetration Test Execution: Post-Exploitation

- Step 5 - 7 of the Kill Chain
- Install software for lateral movement or privilege escalation
- Or as proof that the break-in really happened
- Exfiltrate data (if part of the test objective)
- Or not at all if not mandated by the test objective

Penetration Test Execution: Report

- Two parts
  - Executive summary
    - Target audience: Managers
    - Short recap of objective and activities
    - Found vulnerabilities - structured by severity
    - High level summary of suggestions for improvement
  - Technical report
    - Detailed write-up of activities undertaken
      - Including unsuccessful ones
    - Structured by phases of the penetration test
      - Pre-engagement, information gathering, threat modeling, vulnerability analysis, exploit, post-exploitation
- Get the tone right: It’s not about bragging or embarrassing!

Alternatives/Complements to Vulnerability Assessments

- Checks from outside
  - Websites with checks for specific topics, e. g. Qualys SSL Test
  - Search engines (Google, Shodan, etc.)
  - **Beware: Results are public**
- Cloud services
  - For example: Code scanning/auditing (Coverity, SonarCube, etc.)
  - **Results typically not public, but service (with data) is publicly accessible**
- Checks from inside
  - Shows results for hosts/applications unreachable from the internet
  - Simple/cheap to implement (`nmap`, `OpenVAS`)
  - Vulnerability DB has to be continuously updated
- **Integration into a continuous process is obligatory!**

Penetration Tests vs. Vulnerability Assessments

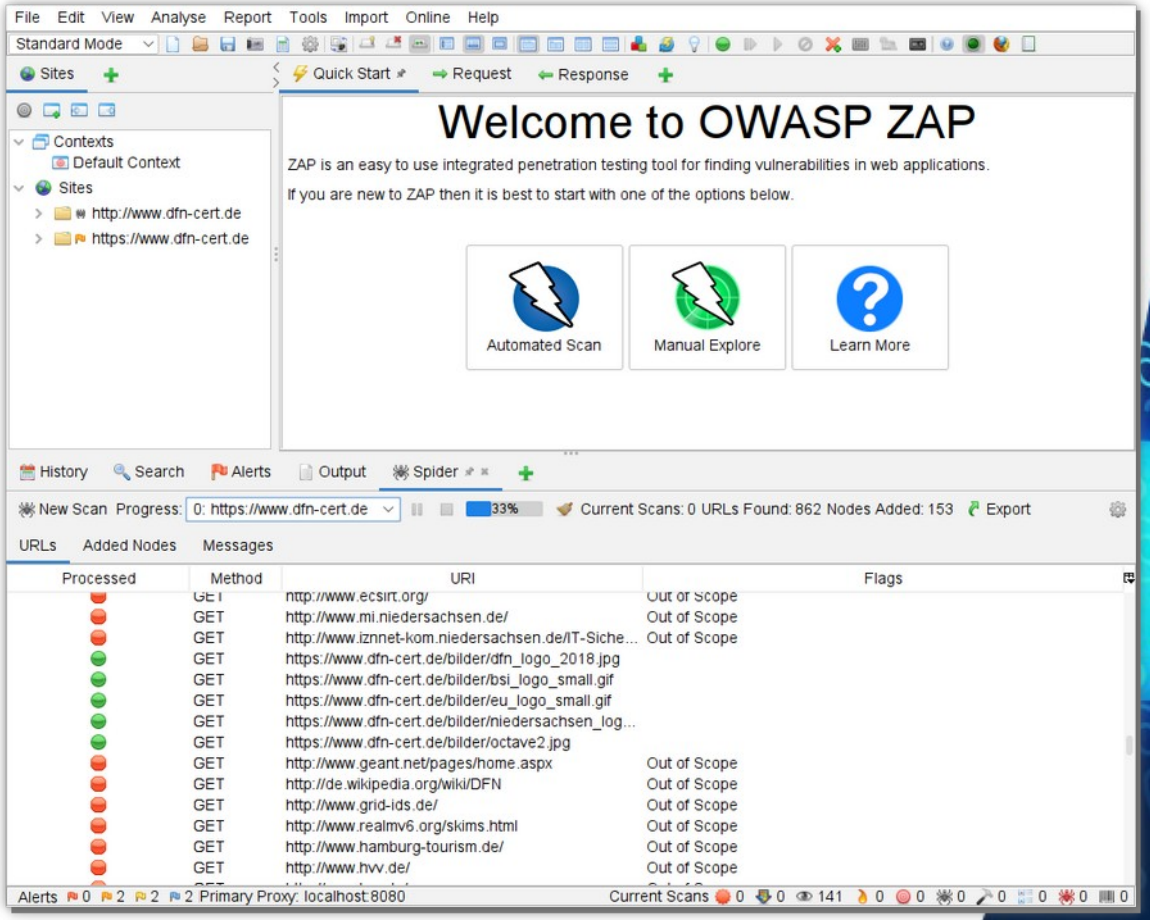
Penetration Tests	Vulnerability Assessments (Scans)
Limited scope (hardware, software, configuration)	Wide scope (e. g. a whole network)
Little predictability (for the “Victim”)	Predictable (Time and place of the test is agreed upon in advance)
Success is precisely defined (goal reached/not reached)	Results are open for debate
Proof-of-Concept for the existence of a vulnerability (on success)	Report with all results and possible improvements
Find at least one vulnerability (and exploit it)	Find as many vulnerabilities as possible (without actually exploiting them)

Tools: Base

- Hardware - a Laptop or even several
  - Special hardware might be required for some penetration tests
    - E. g. for radio penetration tests like WiFi, Bluetooth, LoRaWAN, RFID, etc.
- Operating System - a penetration test distribution of your favorite OS
  - Usually Linux, other OS (OS X, Windows) work just as well
  - Each OS has its strengths and shortcomings
  - Some tools will run only on specific OS
- Hypervisor - choose one of your liking
  - I.e. VMware, VirtualBox, Hyper-V, KVM, etc.
  - For other OS/distributions needed
  - Ready to use Linux VMs (Kali, BackArch, etc.)

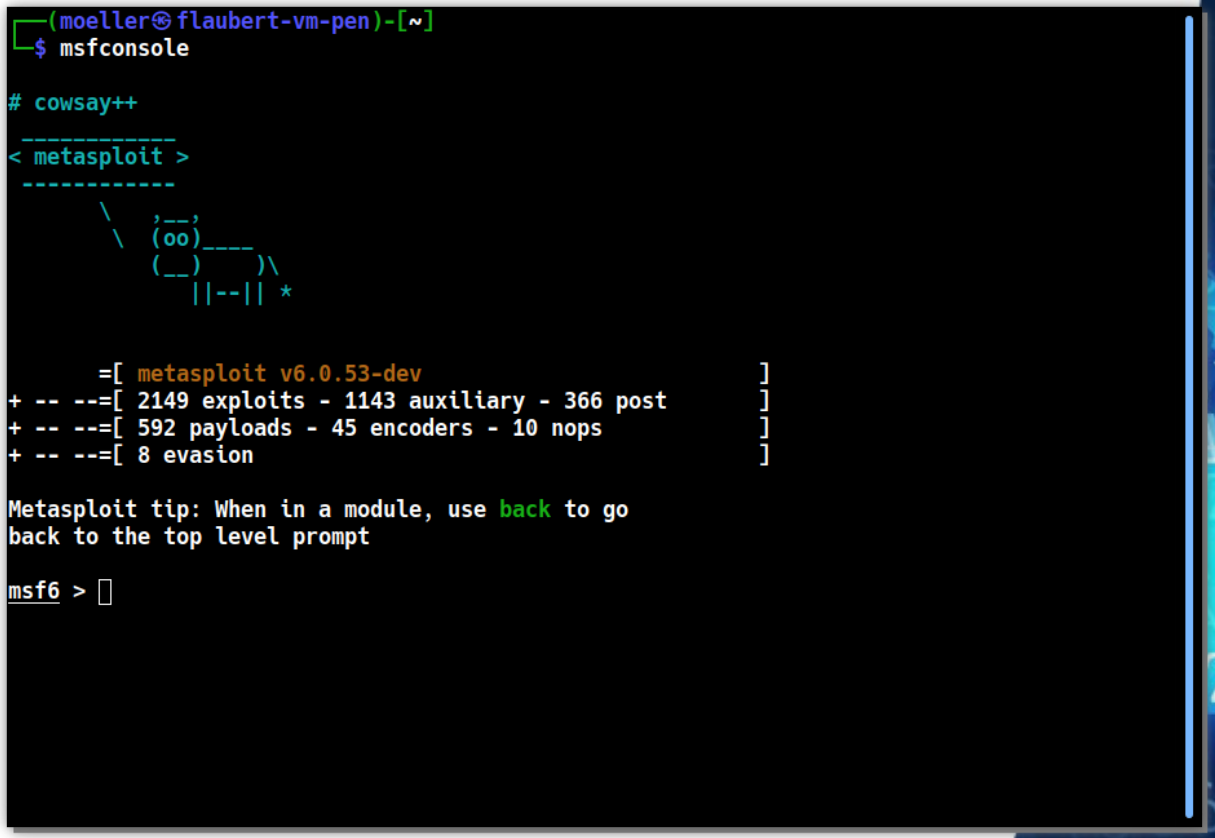
Example 1: OWASP ZAP

- Alias ***O**pen **W**eb **A**pplication **S**ecurity **P**roject **Z**ed **A**ttack **P**roxy*
  - Formerly Paros Proxy
- Man-in-the-middle proxy
  - To examine & manipulate HTTP requests and responses
- Various scanning and spidering functions
  - Can be used as a standalone web-crawler
  - Or interactively while user explores/uses web application



Example 2: Metasploit

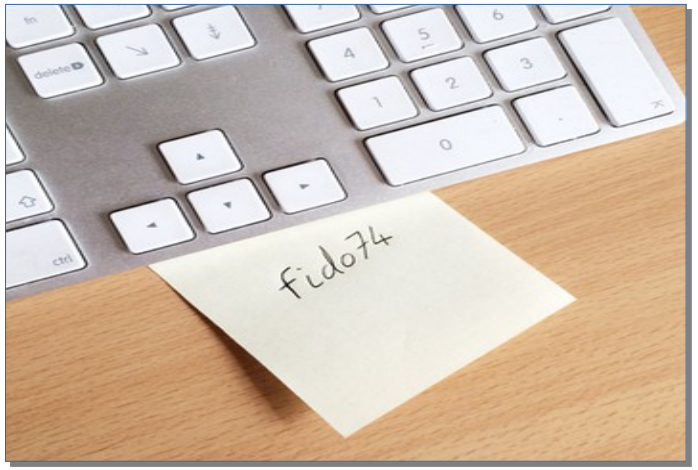
- Several UIs: MsfConsole (CLI), MsfGUI,
- Call/Import data from port-/vulnerability scanners
- Penetration tester can build its own vector/exploit/payload combination
  - Choose components from included ones, import or build from scratch
- Plugins for fuzzing, encoding, scanning, evading IPS, etc.



Further Tests

- Social Engineering
- Even a walk through the offices may reveal vulnerabilities

Password policy?



Source: NASA

Clean desk policy?

Wrap up

- Why undertake penetration tests?
  - Test defense measures, security policies, response plans
  - Find further vulnerabilities/weaknesses after an attack
- Pros & Cons of penetration tests
  - More accurate than simple scans (fewer false positives)
  - More resource-intensive (time, personnel)
- When to do penetration tests
  - Security stance is well-known (i. e. no low-hanging fruits)
  - As part of a continuous integration process (automated tests only)
    - To find new vulnerabilities, or re-appearing old ones (regressions)

Thank you

Any questions?

Next webinar: ***Breach and Attack Simulation,***

15th of September 2021

www.geant.org

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

References:

- Penetration Test Execution Standard (PTES) http://www.pentest-standard.org/index.php
- Wordllists & more: https://github.com/danielmiessler/SecLists
- Michael Bazzel: “Open Source Intelligence Techniques”, 8th Ed., ISBN: 979-8578577086
- https://osintframework.com/
- https://github.com/jivoi/awesome-osint

Sites/Apps to Learn & Practice Penetration Testing

- Hack The Box, https://www.hackthebox.eu/
- bWAPP - buggy Web APPlication, PHP & MySQL, http://www.itsecgames.com/
- HackThisSite, https://hackthissite.org/
- Google Gruyere, Python web app, https://google-gruyere.appspot.com/
- Hellbound Hackers, https://www.hellboundhackers.org/
- OWASP Mutillidae II, https://github.com/webpwnized/mutillidae/
- DVWA - Damn Vulnerable Web Application, PHP & MySQL, https://dvwa.co.uk/
- Defend the Web (formerly: HackThis!!), https://defendtheweb.net/?hackthis
- WebGoat, Java, https://github.com/WebGoat/WebGoat
- Root Me, https://www.root-me.org/?lang=en (default for the site is french)
- Hack Me, https://hack.me/
- OverTheWire, Linux Shell hacking, https://overthewire.org/wargames/
- CTFlearn, https://ctflearn.com/
- DVIAv2 - Damn Vulnerable iOS App, https://github.com/prateek147/DVIA-v2

Penetration Test Distributions

- Linux
  - Kali Linux: https://www.kali.org/
  - BackBox: https://www.backbox.org/
  - Parrot Security Edition: https://parrotsec.org/security-edition/
  - BlackArch: https://blackarch.org/
- Windows
  - PentestBox: https://pentestbox.org/
  - Mandiant Commando VM: https://github.com/fireeye/commando-vm
  - Mandiant Flare VM: https://github.com/fireeye/flare-vm

Penetration Test Tools

- OWASP ZAP: https://owasp.org/www-project-zap/, https://www.zaproxy.org/
- Metasploit: https://github.com/rapid7/metasploit-framework
  - Metasploitable VM: https://github.com/rapid7/metasploitable3