Penetration Testing

IT-Security

Andreas Unterweger

Studiengang Web Business & Technology
FH Kufstein

Sommersemester 2020

Vulnerability Scanning vs. Penetration Testing

- Vulnerability Scanning
  - Automatisierte Suche nach Schwachstellen durch Interne
  - Überprüfter Bereich ist fest definiert
  - Ziel: Aktuellen Sicherheitsstand (z.B. nach Umbau) überprüfen
  - Gefundene Schwachstellen werden **nicht** ausgenutzt
  - Ergebnis: Bericht über gefundene Schwachstellen
- Penetration Testing
  - (Meist) manuelle Suche nach Schwachstellen durch Externe
  - Gesamte Infrastruktur des Auftraggebers im Fokus
  - Ziel: Unbekannte Schwachstellen aufdecken
  - Gefundene Schwachstellen dürfen ausgenutzt werden
  - Ergebnis: Vertraulicher Bericht inkl. erbeuteter Daten an Auftraggeber
- Beide Arten von Überprüfungen durch White Hats

Rechtliche Aspekte von Penetration Testing

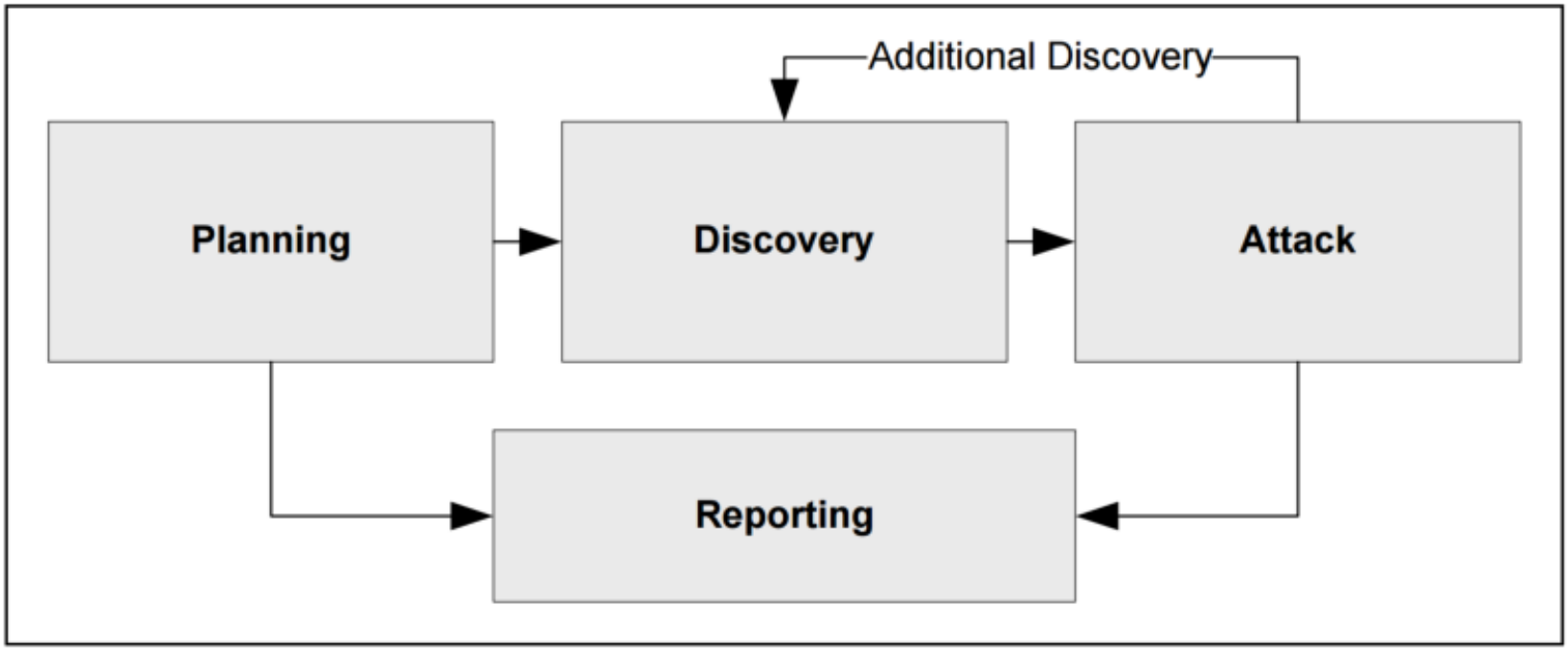
- Penetration Testing simuliert (weitestgehend) echten Angriff
- Angriffe in den meisten Ländern unter Strafe

→ Vertrag oder Vereinbarung notwendig

  - Erlaubt Penetration Testing **explizit**
  - Enthält meist Verschwiegenheitsvereinbarung
  - Definiert unerlaubte Techniken (z.B. DoS)
  - Klärt Haftungsfragen (z.B. bei unbeabsichtigtem DoS)

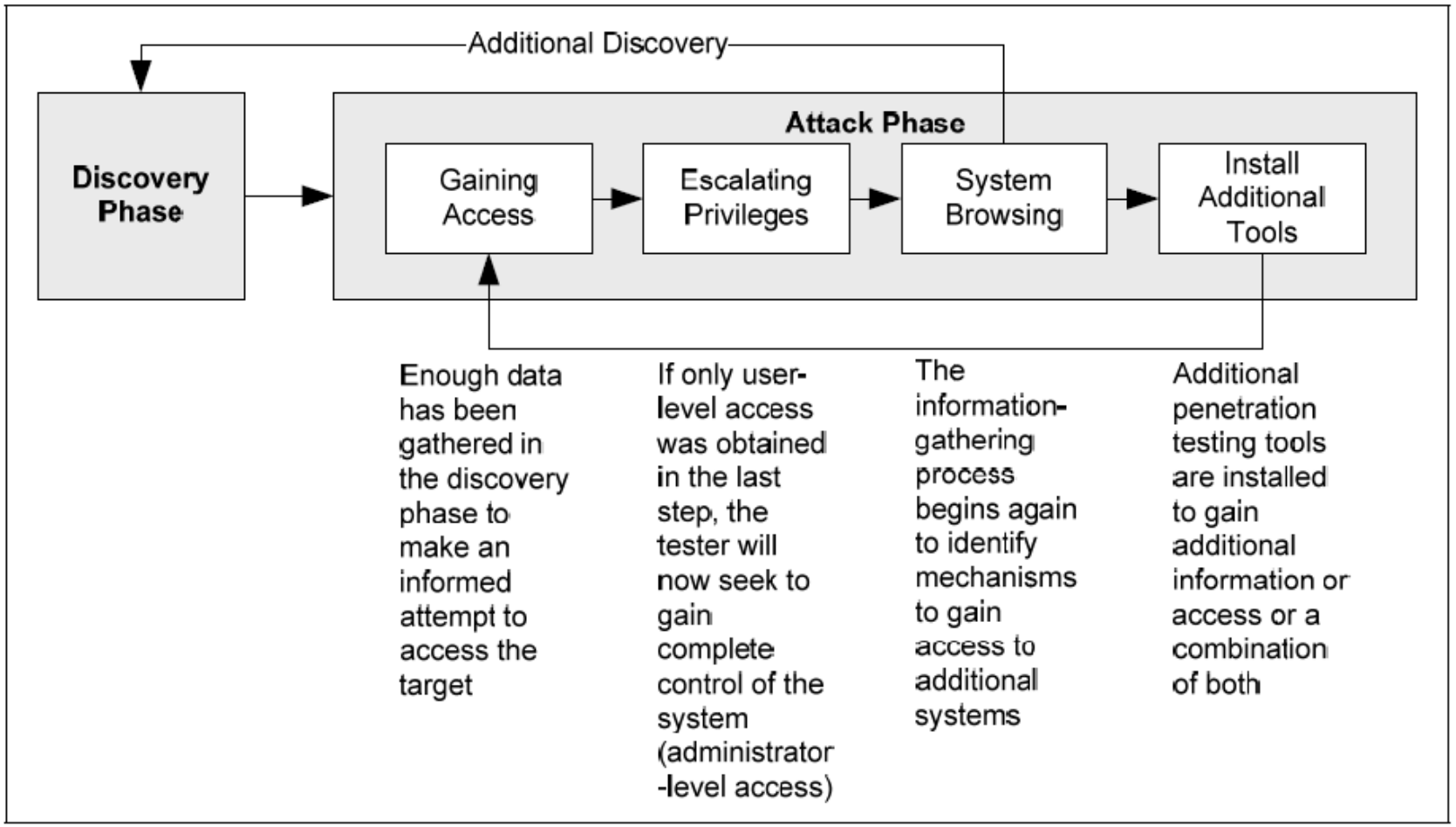
Modell nach NIST 800-115

- NIST 800-115: *Technical Guide to Information Security Testing and Assessment* (Standard)
- Standardmodell für Penetration-Test-Phasen



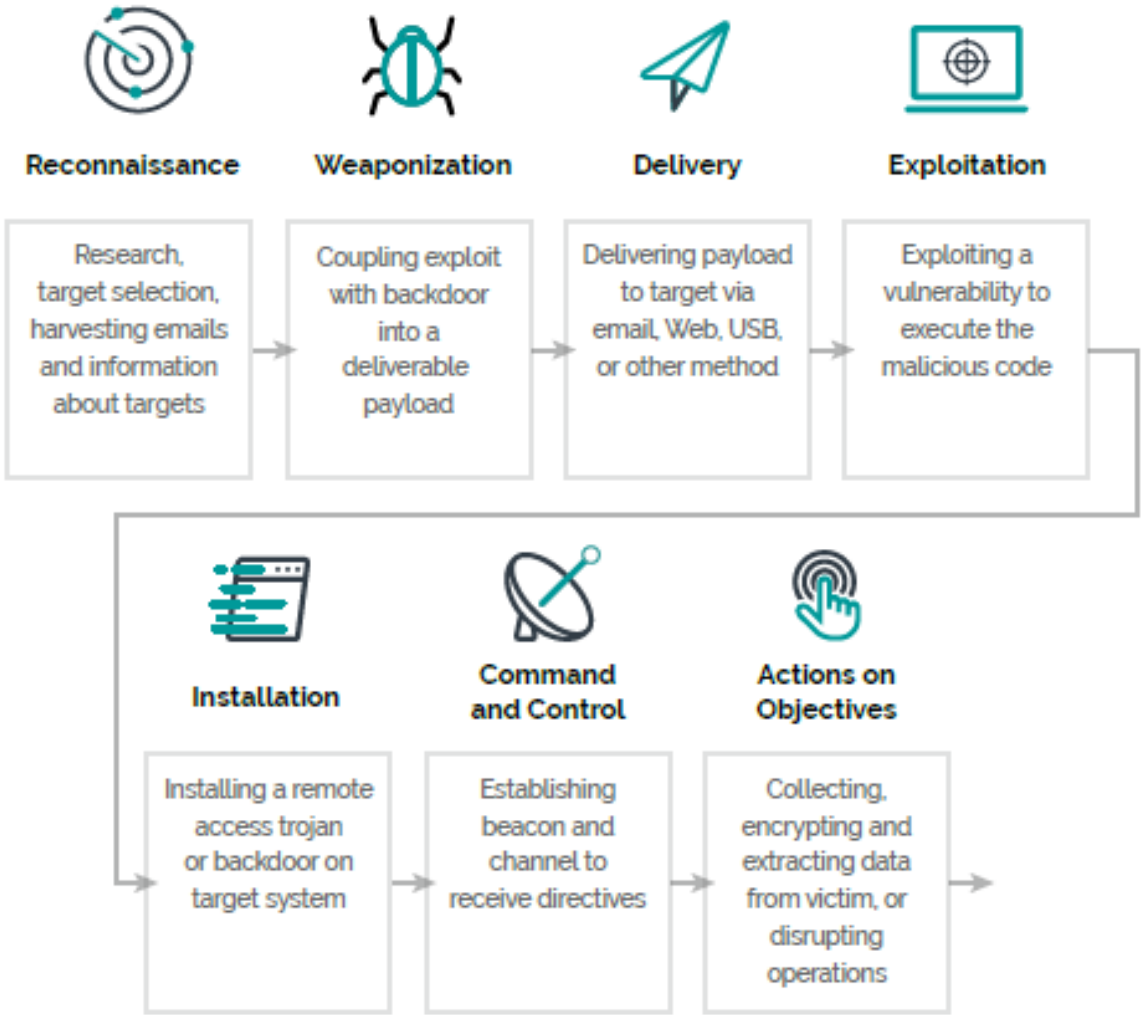
Quelle: Moritz, J.: Pentest Methodologies. `https://www.verifyit.nl/wp/?p=175054` (Zugriff am 11.2.2017), 2016.

Modell nach NIST 800-115 (Detailansicht)



Quelle: CyberQuote: Vulnerability Assessment. `http://www.cqit.sg/vulnerability-assessment/` (Zugriff am 11.2.2017), 2015.

Cyber Kill Chain (typische Schritte eines Angreifers)



Quelle: Johnson, B.: Next-Generation Endpoint Security: Understanding Cyber Attacks from a Hacker's Point-of-View. `https://www.carbonblack.com/2016/04/05/6765/` (Zugriff am 11.2.2017), 2016.

Klassifizierung von Reconnaissance-Strategien

- Aktive Informationsbeschaffung
  - Direkte Kommunikation mit dem Zielsystem bzw. -netzwerk
  - Im Zielsystem direkt sichtbar bzw. erkennbar
  - Vorteil: Detailliertere Informationen
  - Nachteil: Oft leicht zu entdecken und zu blockieren
- Passive Informationsbeschaffung
  - Keine dirkte Kommunikation mit dem Zielsystem bzw. -netzwerk
  - Stattdessen Internet- und/oder Offline-Recherche
  - Vorteil: Recherche bleibt großteils unerkannt
  - Nachteil: Niedrigerer Detailgrad über System(-informationen)

Aktive Informationsbeschaffungsmethoden (Auswahl) I

- `ping` (vgl. letzte Vorlesung)
  - Überprüft Erreichbarkeit des Zielsystems
  - Hauptbedingung: ICMP auf dem System ist nicht deaktiviert
  - Nebenbedingung: ICMP wird nicht gefiltert (z.B. durch Firewall)
- `tracert` (vgl. letzte Vorlesung)
  - Listet Zwischenstationen (z.B. Firewalls) auf
  - Hauptbedingung: Zwischenstationen unterstützen ICMP
  - Nebenbedingung: ICMP wird nicht auf dem Weg gefiltert
- Informationen über DNS (`nslookup` bzw. `dig`)
  - Hosts über DNS-Records ausfinding machen
  - IP-Adressen über Reverse DNS ermitteln
- Geo-Location (Standort aus IP-Adresse ermitteln)
  - Zugewiesene IP-Adressbereiche sind meist geografisch zugeordnet
  - Beispiel: `http://www.ip-adress.com/ip_tracer/8.8.8.8`

Aktive Informationsbeschaffungsmethoden (Auswahl) II

- Hosterkennung (engl. *host discovery*)
  - Ping an alle Hosts im Subnetz (sog. *ping sweep*)
  - Hosts, die online sind und ICMP unterstützen, antworten
  - Funktioniert nicht, wenn Firewall ICMP blockiert
  - Alternative: Versuch, TCP- oder UDP-Verbindung aufzubauen
- Port Scanning
  - Versuch, offene TCP- oder UDP-Ports zu finden (Antwort vom Host)
  - Bekannte Portnummern erlauben Rückschlüsse auf Dienste, z.B.

22	SSH (TCP, UDP)	110	POP3 (TCP)
25	SMTP (TCP)	143	IMAP (TCP)
53	DNS (TCP, UDP)	*443*	*HTTPS (TCP)*
80	HTTP (TCP)	*993*	*IMAPS (TCP)*

  - Geschlossene Ports: Antwort per ICMP, dass Dienst nicht verfügbar ist
  - Gefilterte Ports: Keine Antwort (z.B. durch Firewall)

Aktive Informationsbeschaffungsmethoden (Auswahl) III

- Diensterkennung (engl. *service detection*)
  - Versuch herauszufinden, welcher Dienst an offenem Port lauscht
  - *Banner Grabbing*: Dienstname, Versionsnummer oder andere Identifikation aus Antwort extrahieren
  - Standard-Banner-Grabbing-Werkzeuge für häufig verwendete Dienste
  - Erfordert je nach Protokoll manchmal manuelle Protokollanalyse
- Betriebssystemerkennung (engl. *Operating System (OS) detection*)
  - Verschiedene Betriebssysteme bzw. -versionen antworten teilweise unterschiedlich auf bestimmte Pakete (ohne Details)
  - Muster erlauben Rückschlüsse auf Betriebssystem(-version)
  - Werkzeuge mit Datenbanken bekannter „Fingerabdrücke“
  - Oft leicht erkennbar durch Einbruchserkennungssystem (engl. *Intrusion Detection System* – IDS) $\rightarrow$ Abbruch durch Verbindungsblockade

Schwachstellenermittlung (engl. *vulnerability scanning*)

- Suche nach Schwachstellen auf Basis der
  - Laufenden Dienste
  - Installierten Software
  - Installierten Betriebssysteme
  - Versionsinformation (Patch-Status)
  - Konfigurationsinformation (z.B. Verschlüsselungsunterstützung)
  - ...
- Suchoptionen: Automatisch oder manuell
  - Vulnerability Scanner (mit Datenbank) direkt auf Zielsystem angewandt
  - Exploit Database (`http://www.exploit-db.com`)
  - Common Vulnerabilities and Exposures (`http://cve.mitre.org/`)
  - ...
  - Metasuchmaschinen, z.B. `http://exploitsearch.com/`

Typische Angriffsziele

- Unterscheidung nach Gerätetyp
  - Server: tendenziell schwieriger (technisch versierte Benutzer)
  - **Client**: tendenziell einfacher (oft unbedarfte Benutzer)
- Ausnutzung von Schwachstellen in
  - **Applikationen**
  - Diensten (nur Server)
  - **Betriebssystem**
- Menschliche Komponente oft entscheidend (Social Engineering)

Typische clientseitige Angriffsmethoden (Auswahl)

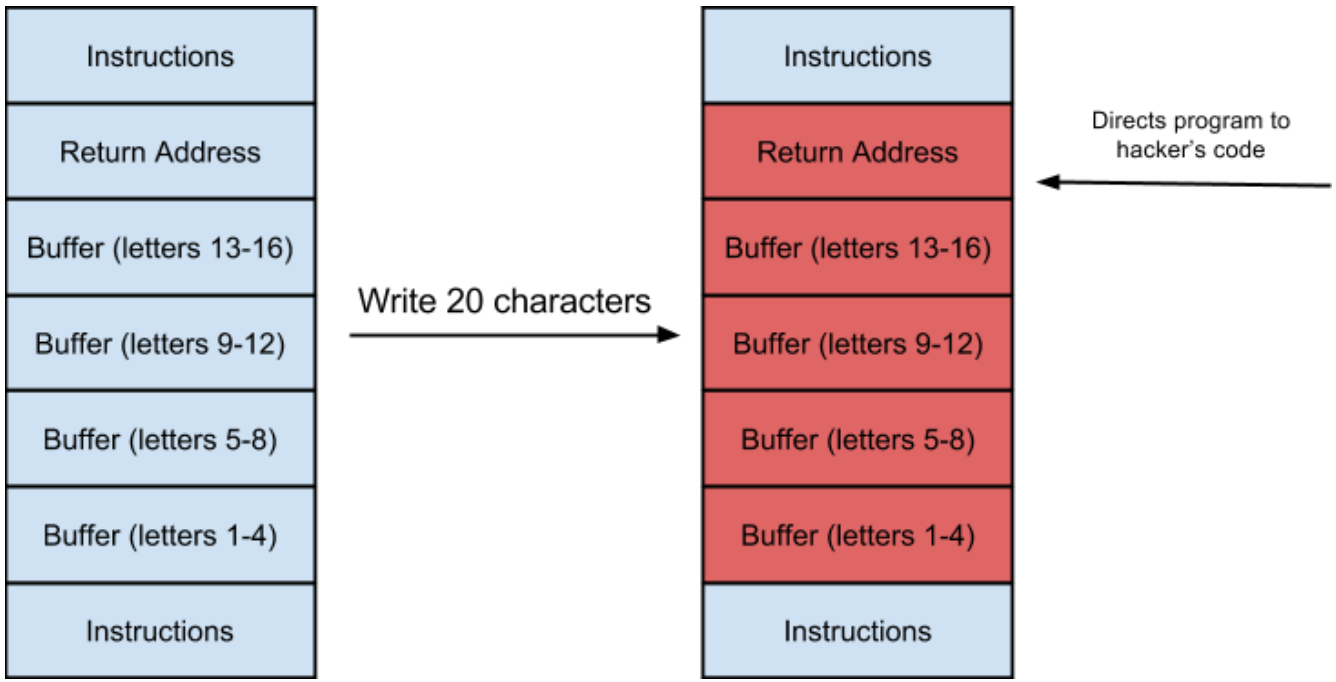
- Email (mit Anhang)
  - Installation von Schadsoftware
  - Öffnen des Anhanges bzw. der Email meist erforderlich
  - Ausnutzung von Sicherheitslücken in Email- bzw. Anwendungsprogramm (z.B. PDF Reader, Office)
- Besuch von Webseite (z.B. aus Email mit Link)
  - Installation von Schadsoftware (z.B. *Drive-By-Download*)
  - Bei Links Anklicken meist erforderlich
  - Ausnutzung von Sicherheitslücken im Browser oder dessen Plugins
- Gemeinsamkeiten der genannten Methoden
  - Menschliches Zutun notwendig
  - Erfolgswahrscheinlichkeit durch Social Engineering höher
  - Erfolgswahrscheinlichkeit bei veraltetem Patchstatus höher

Typische Angriffstechniken (Auswahl) I

- Pufferüberlauf (engl. *buffer overflow*)
  - Puffer auf Stack über Maximum gefüllt (meist mangels Überprüfung)

  → Rücksprungadresse wird überschrieben

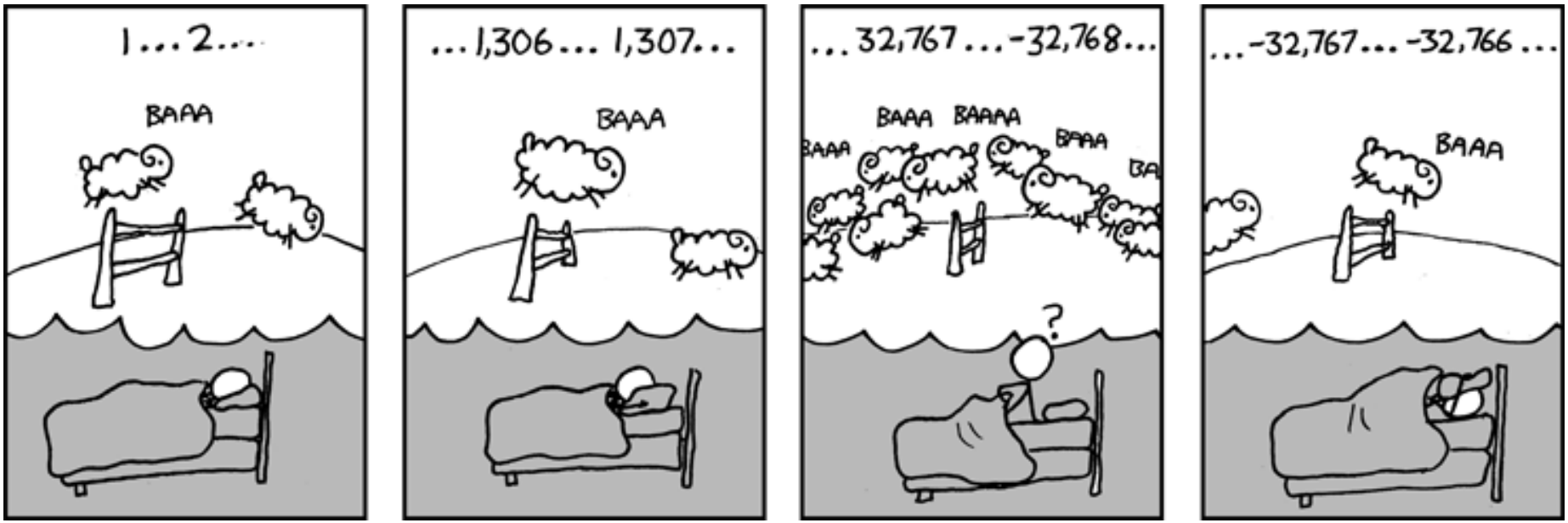
  → Rücksprungadresse von Pufferinhalt abhängig
  - Puffer kann Schadcode enthalten → Ausführung möglich



Adaptiert von sever408: Canary (buffer overflow). `http://www.cbi.umn.edu/securitywiki/CBI_ComputerSecurity/MechanismCanary.html` (Zugriff am 12.2.2017), 2015.

Typische Angriffstechniken (Auswahl) II

- Ganzzahlüberlauf (engl. *integer overflow*)
  - Ganzzahltypen sind größenbegrenzt
  - Überschreitung der Obergrenze bei Berechnungen führt zu Überlauf
  - Wenn Berechnungsergebnis (Speicher-)Größenangabe $\rightarrow$ Pufferüberlauf



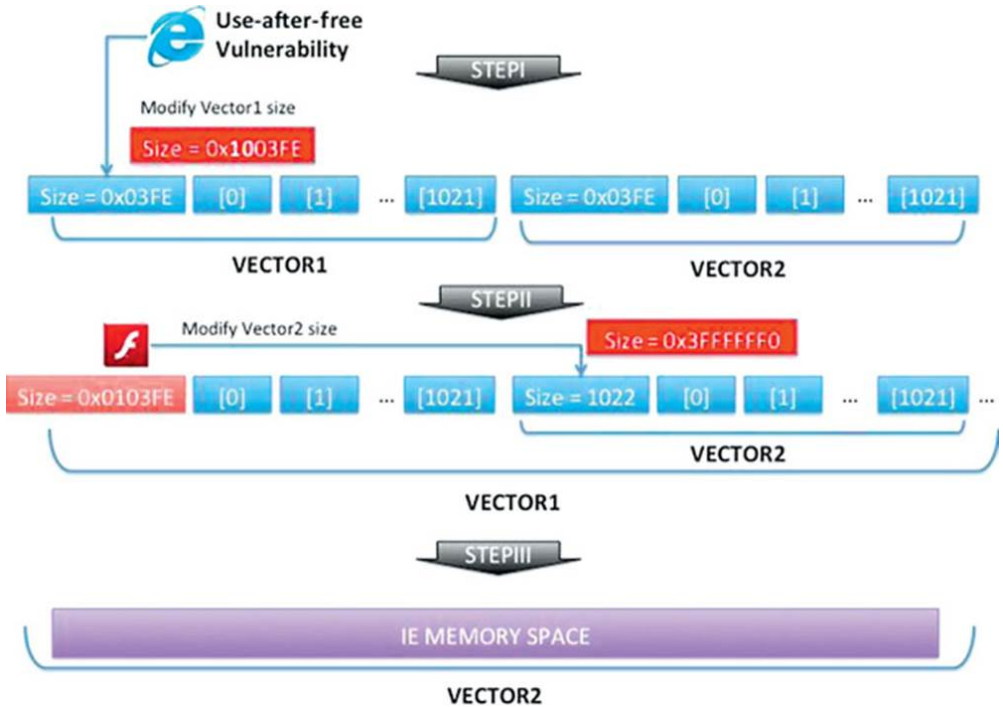
Quelle: Munroe, R.: Can't sleep. `https://xkcd.com/571/` (Zugriff am 12.2.2017), unbekannt.

Typische Angriffstechniken (Auswahl) III

- *Arbitrary code execution / Remote code execution*
  - *Heap spraying*: Wie Pufferüberlauf, aber auf Heap (statt auf Stack)

  → Datenbereiche eines Prozesses (z.B. Längen) werden überschrieben
  - Überschriebene Teile enthalten Schadcode

  → Kontrolle über Prozess → Ausführung von beliebigem Code



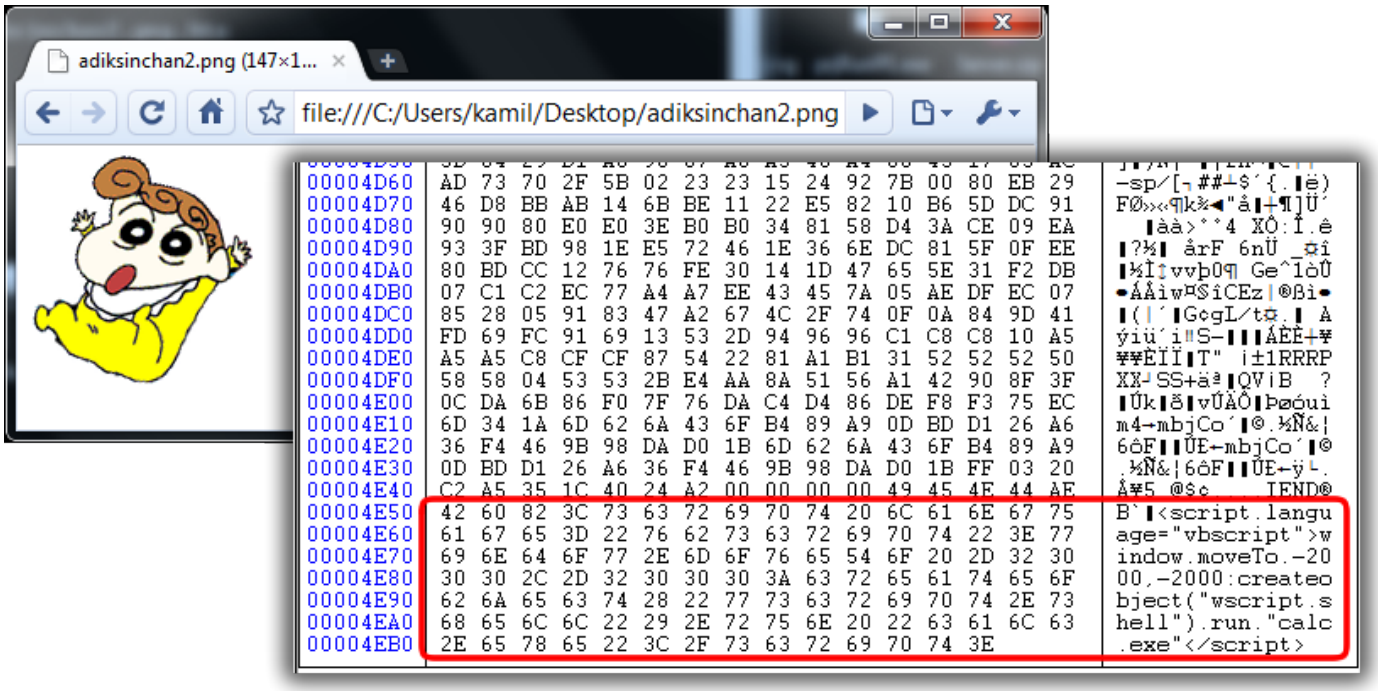
Quelle: Unbekannt: How to Defeat Advanced Malware: New Tools for Protection and Forensics (2015). `http://apprize.info/security/malware/2.html` (Zugriff am 12.2.2017), 2015.

Typische Angriffstechniken (Auswahl) IV

- Beispiel für *Arbitrary code execution* über falsche Dateiendung
  - Beispiel: Daten nach Bilddaten (am Dateiende)

  → Bild bleibt gültig (Daten nach Bilddaten werden ignoriert)
  - Ändern der Dateiendung ändert Interpretation

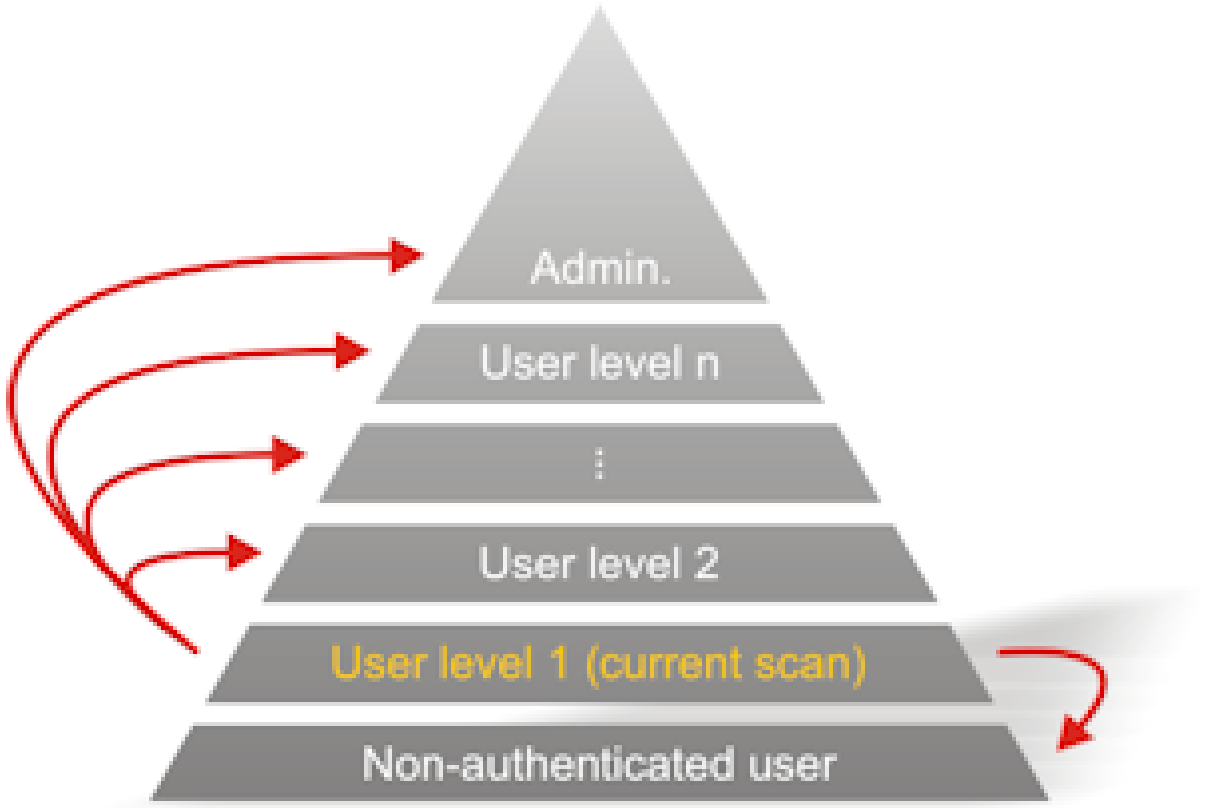
  → Bilddaten werden ignoriert → Ausführung von beliebigem Code



Quelle: Alta, K.: Embedded Script on Images file allowed Arbitrary Code Execution. `http://blog.data0.net/2010/08/embedded-script-on-images-file-allowed.html` (Zugriff am 12.2.2017), 2010.

Typische Angriffstechniken (Auswahl) V

- Rechteausweitung (engl. *privilege escalation*)
  - Ziel: Höhere Rechte bzw. Rechte eines anderen Nutzers erlangen
  - Erweiterte Rechte ermöglichen erweiterten Zugriff



Quelle: IBM: Privilege Escalation view. `https://www.ibm.com/support/knowledgecenter/en/SSPH29_9.0.3/com.ibm.help.common.infocenter.aps/t_TestPrivilegeEscalation080.html` (Zugriff am 12.2.2017), 2015.

Überblick: Nach dem Angriff

- Statuserhebung
  - Welche Rechte (keine, Benutzer, Admin) hat man?
  - Welche Informationen sind direkt zugreifbar?
  - Welche neuen (Netzwerk-)Verbindungen gibt es?
- Weitere Reconaissance
  - Andere Benutzer auf dem Rechner (für Rechteausweitung)
  - **Daten auf dem Rechner**
  - Andere Rechner im Netzwerk
- Spuren beseitigen (z.B. Logdateien)
- **Zugriff aufrecht erhalten**
  - Installation einer Hintertür
  - Passwörter sammeln und versuchen zu knacken
- Dokumentieren (**Bericht schreiben**)

Datensammlung

- Betriebssysteminformationen (für weitere Angriffe)
- Anwendungs(-versions-)informationen (für weitere Angriffe)
- Passwörter (vgl. später)
- Live-Aufnahmen
  - Bildschirm (Screenshots)
  - Tastatur (über temporären Keylogger)
  - Umgebung, z.B. über Webcam oder Mikrofon
- Benutzerdateien
- Dateien auf Netzlaufwerken
- ...

→ Zusätzliche Informationen über neue Angriffsziele

Hintertüren

- Vermeiden neuerlicher Ausnutzung derselben Angriffsvektoren
- Installation von Schadsoftware (Varianten vgl. letzte Vorlesung)
- Beispiel: Netzwerkdienst, der lauscht und auf Befehle wartet
- Vor Installation
  - Deaktivierung von Virenscannern
  - Deaktivierung von Firewalls
  - Deaktivierung anderer Sicherheitsmaßnahmen
- Installation im System möglichst dauerhaft
- Installation für automatischen Start, z.B. in Registry (Windows)
- Automatische Verbindung zu Angreifer möglich

Passwortsammlung

- Passwörter müssen (z.B. unumkehrbar) gespeichert werden

→ Zugriff auf Passwortinformationen mit entsprechenden Rechten

- Sammlung und Übertragung von Passwortinformationen ermöglicht
  - Knacken von Passwörtern **offline** (unauffälliger)
  - Finden von Mustern (je nach Speicherart) → gleiche Passwörter
  - Unauffällige Rechteausweitung bei Erfolg („normaler“ Login)
- Passwortkomplexität ist Sicherheitsfaktor (spätere Vorlesungen)

Quelle: G: [01.14] All Password List, Dictionary collection for various purposes. `http://appnee.com/all-password-list-dictionary-collection-for-various-purposes/` (Zugriff am 12.2.2017), 2017.

Penetration-Test-Bericht

- Kriterien
  - Beinhaltet detaillierte Ursachenforschung
  - Beinhaltet **keine** nicht ausnutzbaren Schwachstellen
  - Einfach, verständlich und ortografisch korrekt
  - Konsistent formatiert und strukturiert (Abschnitte)
- Zielgruppengerechter Fokus
  - Geschäftsführungsebene: Kurz und prägnant mit Risikobewertung (vor vs. nach Gegenmaßnahmen)
  - Führungsebene (mittleres Management): Schwachstellenbewertung und allgemeine Empfehlungen zur Erhöhung der Sicherheit
  - Techniker: Details zur Schwachstellenauffindung, -ausnutzung und -beseitigung
- Beispiel: `https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf`

Danke für die Aufmerksamkeit!

Fragen?