Automated Success Verification of Exploits for Penetration Testing with Metasploit

CODE 2020 | 11.11.2020 | Tobias Appel

Vorstellung Sprecher

- Wissenschaftlicher Mitarbeiter am Leibniz Rechenzentrum
- Mitglied des Munich Network Management Team
- IT Security Consultant & Trainer

- Forschungsschwerpunkte:
  - IT Security Operations (SIEM / IDS)
  - Automated Penetration Testing
  - Vulnerability Management

- Zertifikate
  - ICO ITSec Penetration Tester
  - ICO ISMS Security Officer (ISO 27001)
  - …

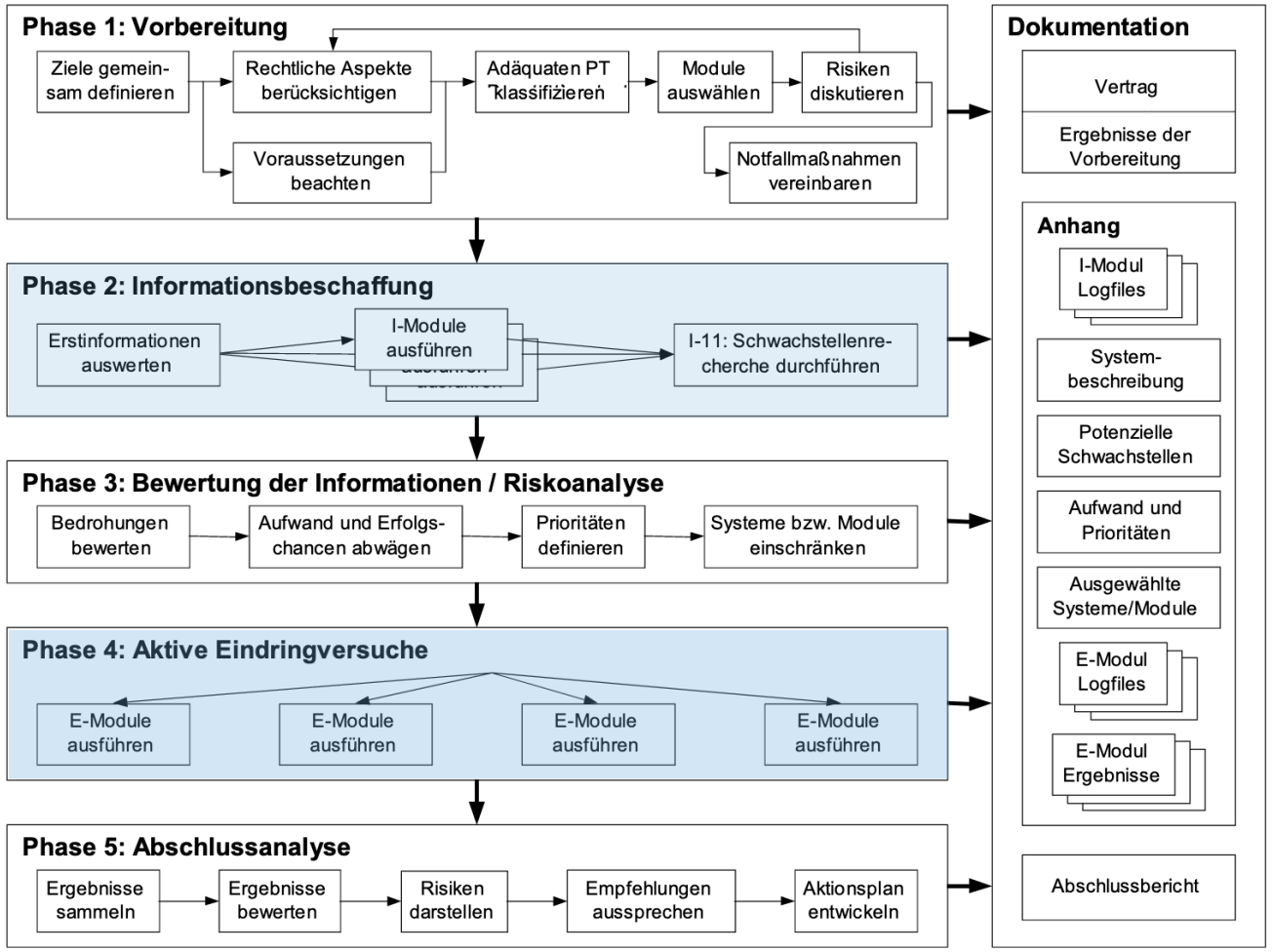
Durchführungskonzept für Penetrationstests des BSI

Phase 2: Informationsbeschaffung

→ z.B. Schwachstellenscan

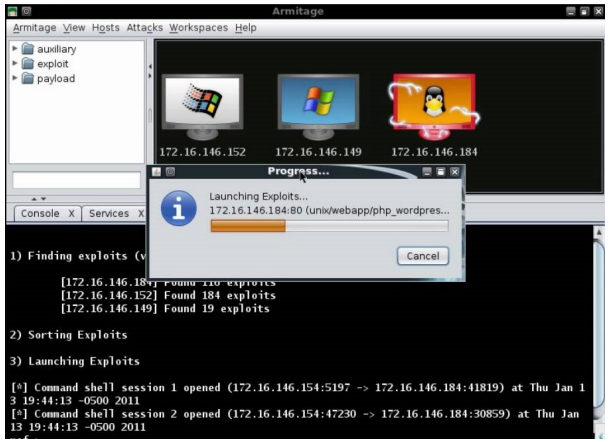
Phase 4: Aktive Eindringversuche

→ z.B. Exploits mit Metasploit

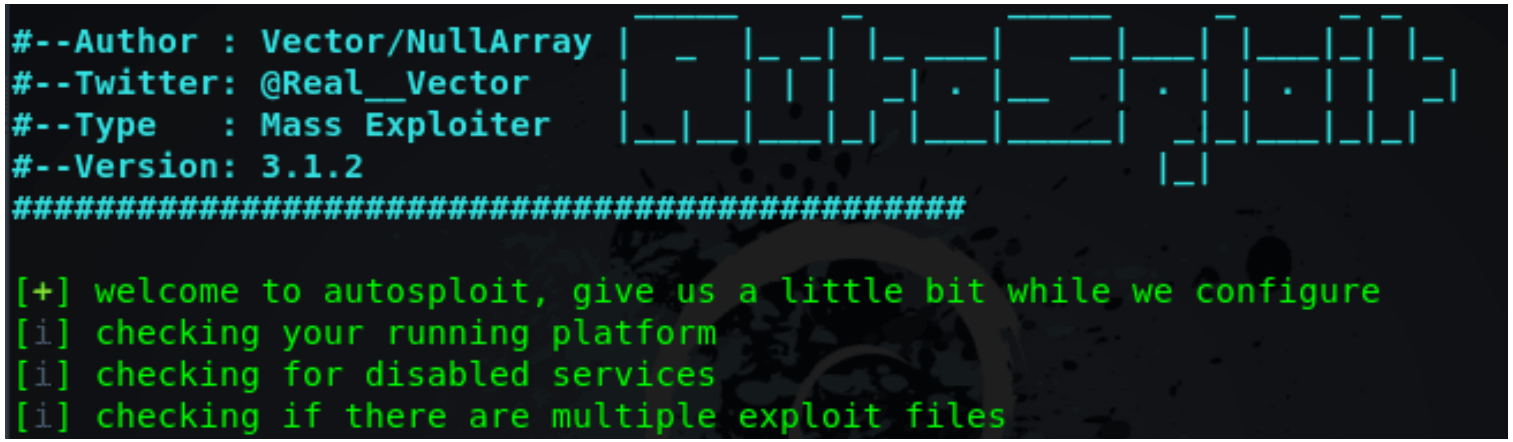


Quelle: BSI[1]

Bekannte Exploiting Tools können nur bedingt unterstützen



Hail Mary (db_autopwn)[2]



AutoSploit[3]

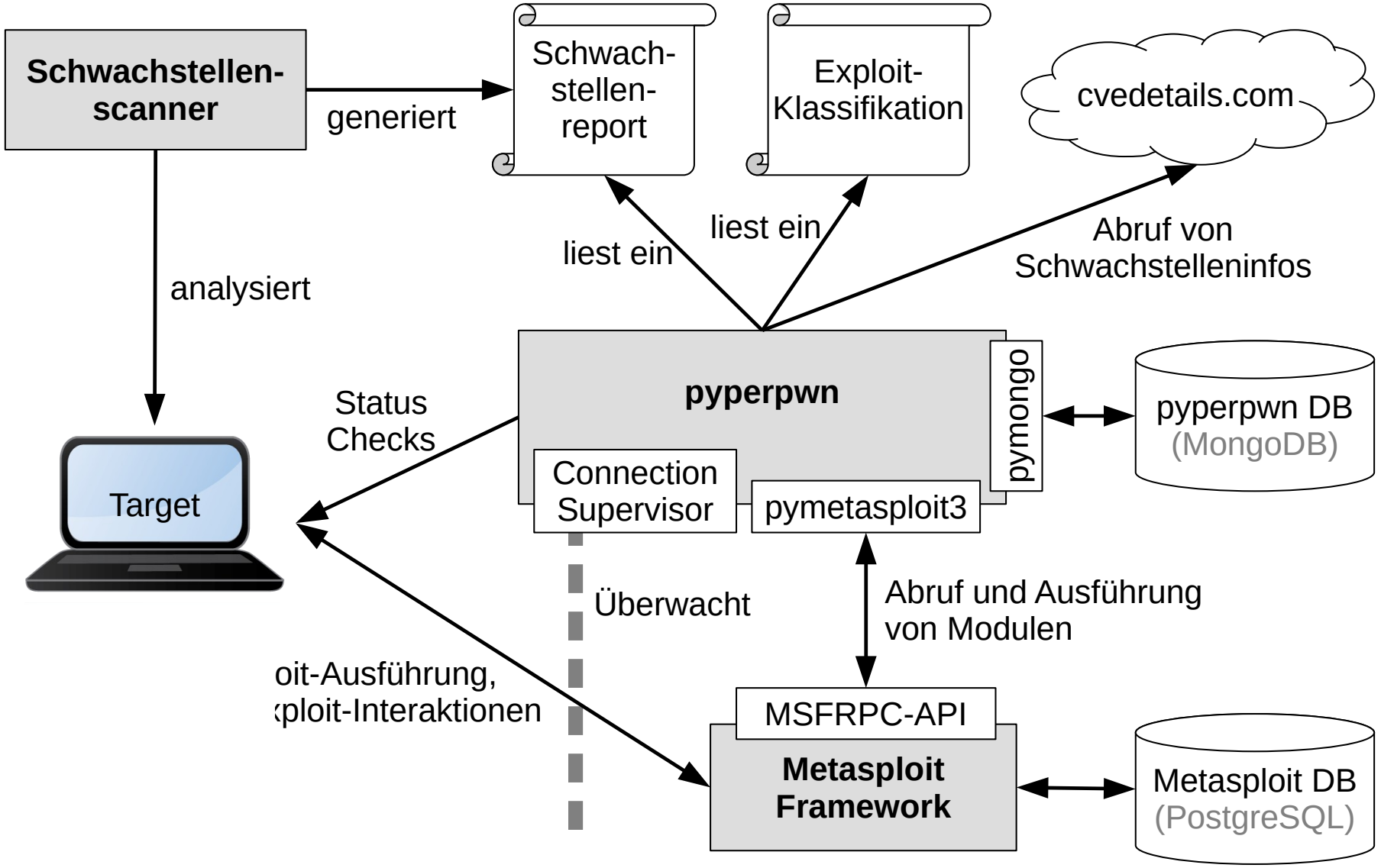
- **Mangelhafte Ziel-Analyse**
- **Ungenaue Exploit-Ausführung**
- **Ungenügende Erfolgsbewertung**

pyperpwn[4] to the rescue

- Entwickelt mit Python3 und pymetasploit3

- Für (teil-)automatischen Penetrationstest entworfen
- Umfangreiches Reporting und Erfolgsbewertung einzelner Exploits
- Kann überprüfen, ob Sicherheitslücken geschlossen worden sind

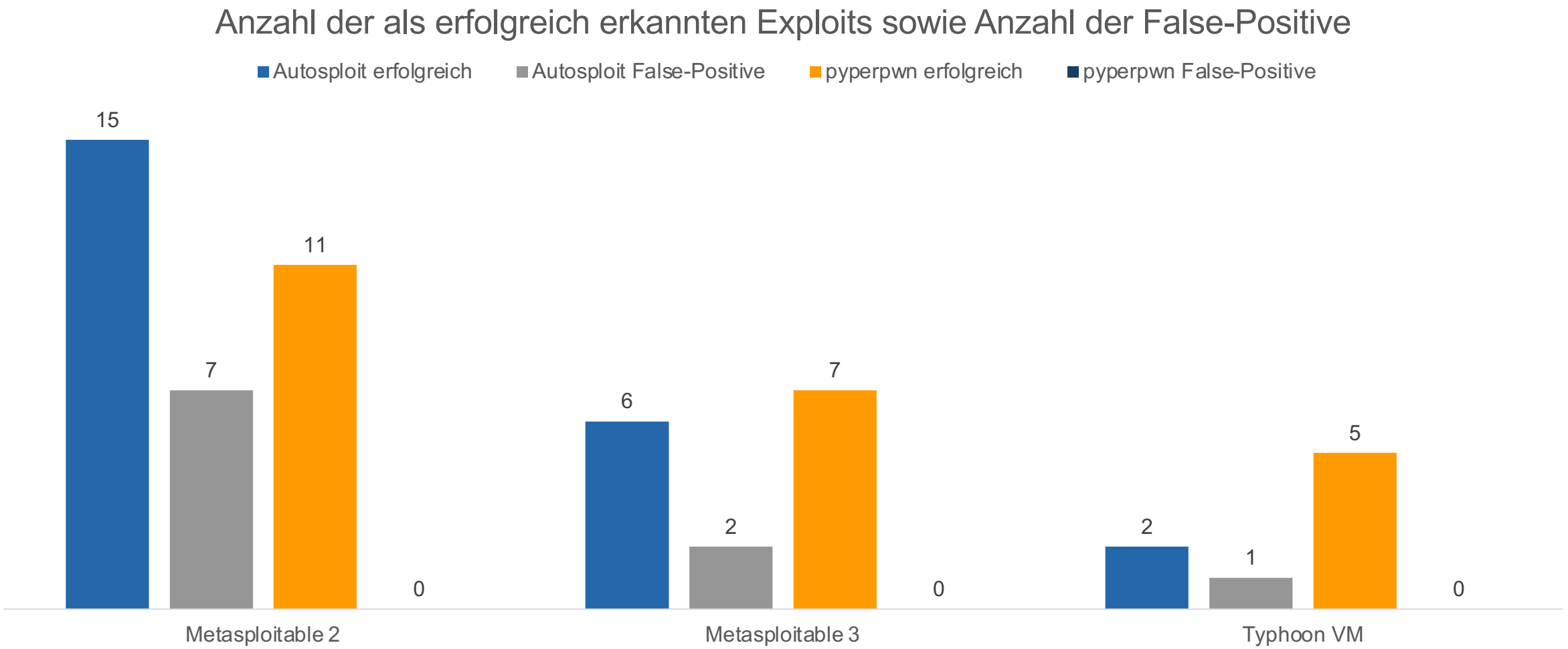
Architektur und Schnittstellen



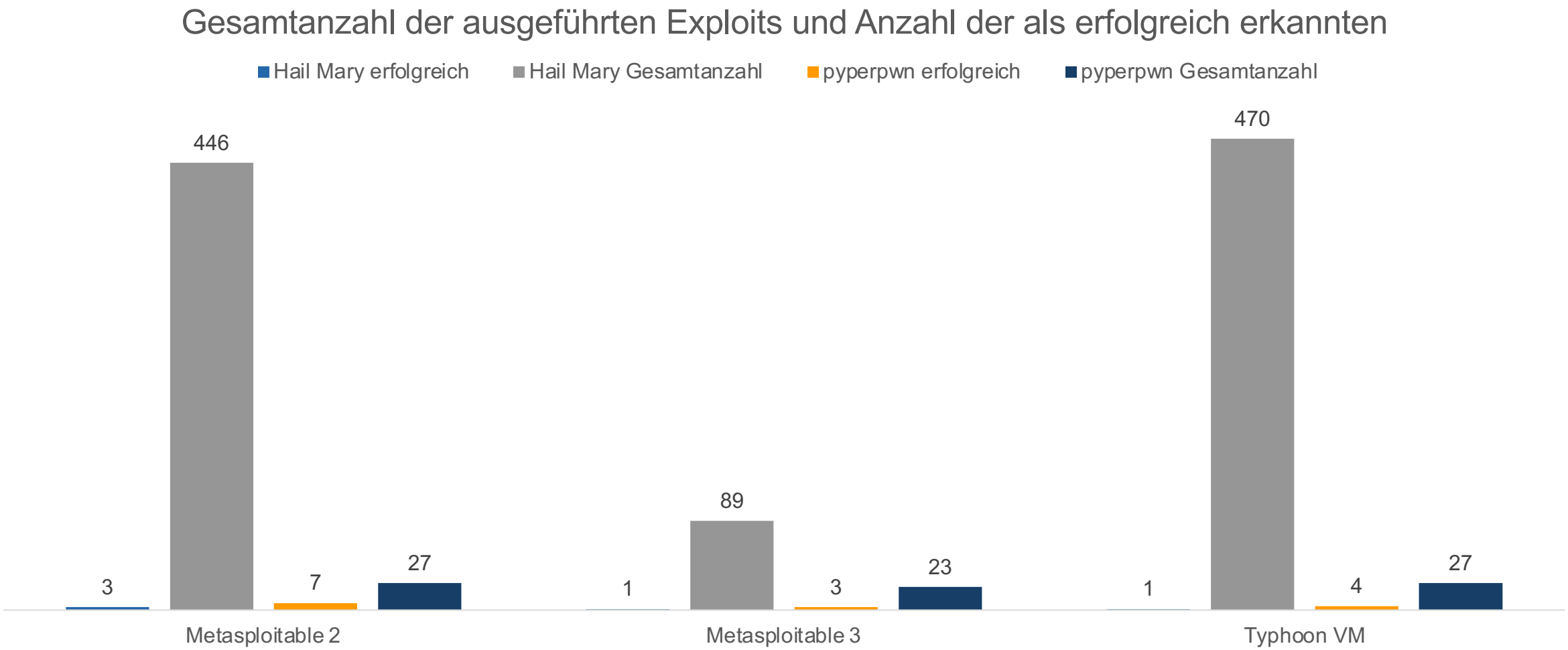
Anschließend erhalten wir sehr detailierte Reports (CSV-Format)

Spaltenname	Beschreibung	Beispiel
Exploit	Verwendeter Exploit in Metasploit	**multi/http/tomcat_mgr_upload**
Success	Angabe ob Exploit erfolgreich war	**True**
Execution Result	Ergebnis der Ausführung	**EXPLOIT_EXECUTED_GAINED_ACCESS**
Execution Impact	Detektierte Auswirkung	**{'ping': <ServiceStatus.UP_ACCESSIBLE: 0>, 'nmap': <PortStatus.OPEN: 0>, 'http': <ServiceStatus.UP_ACCESSIBLE: 0>}**
Session Info	Output der Remote Session	**{'info': 'tomcat55 @ metasploitable', 'meterpreter: getuid': 'Server username: tomcat55', …. }**
…	…	…

Im Gegensatz zu Autosploit hat pyperpwn keine False-Positives[5]



... und benötigt deutlich weniger Exploits als Hail Mary[5]



Fazit und Ausblick

- Ergebnis
  - Deutlich effizienter und effektiver als bisherige Exploiting-Tools
  - Unterstützt Penetrationtester bei der Arbeit
  - Auf dem Weg zum “Fully Automated Penetration Testing” in Zukunft

- Einschränkungen
  - Bislang nur remote ausführbare Exploits (keine lokalen Exploits)
  - Keine Verkettung von Exploits (Exploit-Chain)

- Erweiterungsmöglichkeiten
  - Unterstützung für Exploit-Chain
  - Einbettung in Test-Umgebung[6] um Integrität des Ziels zu verifizieren

Code bei github:

https://github.com/dial25sd/pyperpwn

Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kontakt

Leibniz Rechenzentrum, Boltzmannstr. 1, 85748 Garching
Tobias.Appel@lrz.de

Quellen

- [1] Durchführungskonzept für Penetrationstests, BSI 2020, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Penetrationstest/penetrationstest.pdf?__blob=publicationFile&v=3
- [2] AutoSploit: https://github.com/NullArray/AutoSploit
- [3] Hail Mary (db_autopwn): https://blog.cobaltstrike.com/2013/07/17/the-origin-of-armitages-hail-mary-mass-exploitation-feature/
- [4] pyperpwn Code: https://github.com/dial25sd/pyperpwn
- [5] Automated Success Verification of Exploits for Penetration Testing with Metasploit, Sedlmeir S., LMU 2019, http://www.mnm-team.org/pub/Fopras/sedl19/
- [6] Test-Umgebung zur Evaluierung von Schwachstellenscannern, Würz R., LMU 2020, http://www.mnm-team.org/pub/Diplomarbeiten/wuer20/