2009

COLLABORATIVE PENETRATION TESTING

Severin Winkler
Secure Business Austria

Christian Proschinger
Raiffeisen Informatik

COLLABORATIVE PENETRATION TESTING

Severin Winkler[1], Christian Proschinger[2]

Kurzfassung
Penetrationstests stellen im IT-Sicherheitsmanagement großer Unternehmen einen Fixpunkt bei technischen Audits dar. Die Durchführung eines solchen Audits mit den Mitteln eines realen Angreifers stellt einen besonderen Mehrwert dar, da es eine realistische Einschätzung der aktuellen Sicherheitssituation zulässt. Collaborative Penetration Testing beschreibt die Möglichkeit der effektiveren Durchführung solcher Audits, um mit der Professionalisierung im Cybercrime Bereich Schritt zu halten. Durch Spezialisierung und daraus resultierende Arbeitsteilung wird ein effizienterer Einsatz des vorhandenen Testbudgets erreicht. Vorhandene Vorgehensmodelle können so in ihrer Produktivität gesteigert werden. Unseren Ansatz stellen wir anhand eines Prototyps vor.

1. Einführung

Ein kollaborativer Penetrationstest (Collaborative Penetration Test) ist ein von mehreren, in der Regel zeitlich und physisch getrennt arbeitenden Menschen durchgeführter Test. Dadurch ergeben sich einerseits neue Möglichkeiten, andererseits werden Fragen und Probleme zur Arbeitsweise der Gruppe und dem allgemeinen Ablauf aufgeworfen. Es handelt sich dabei um eine Kombination aus den Forschungsgebieten des Penetration Testings[3] sowie der Computer Supported Collaborative Work (CSCW) [1] [18].

Der Vorteil eines kollaborativen Penetrationstests gegenüber einem nicht kollaborativ durchgeführten Audit liegt im Nutzengewinn, erreicht durch Spezialisierung und gezielte Werkzeugnutzung. Ziel des vorliegenden Beitrages ist es, die Voraussetzungen für kollaboratives Penetration Testing aufzuzeigen und ein anhand dieser Anforderungen entwickeltes Tool vorzustellen.

Als Penetrationstest wird im Allgemeinen ein Sicherheitstest eines IT Systems bezeichnet, bei dem die Tester Mittel und Methoden nutzen, die auch von realen Angreifern verwendet werden [2]. Im Gegensatz zu einem automatisierten Schwachstellenscan werden gefundene Sicherheitslücken in Absprache mit dem Auftraggeber auch tatsächlich ausgenutzt. Auf diese Weise kann Schritt für Schritt immer tiefer in ein System eingedrungen werden, wodurch falsch-positive Ergebnisse ausgeschlossen bzw. weitere Schwachstellen aufgedeckt werden können.

1 Secure Business Austria, A-1040 Wien, Favoritenstraße 16.

2 Raiffeisen Informatik GmbH, A-1020 Wien, Lilienbrunngasse 7-9.

Des Weiteren wird diese Art eines technischen Sicherheitsaudits durch folgende Punkte charakterisiert:

- Ort, Zeitpunkt und mögliche Zielsysteme sind mit dem Auftraggeber (z.B. einem Systembetreiber oder Systemeigentümer) abgestimmt.
- Der Angriff kann bei Beeinträchtigungen der getesteten Umgebung jederzeit abgebrochen werden.
- Ein festgelegtes Budget begrenzt den Zeitrahmen.
- Die gefundenen Schwachstellen werden bewertet und in einem Bericht präsentiert.
- Eine manuelle Prüfung von automatisiert generierten Scanergebnissen wird durchgeführt.

Aufgrund der zahlreichen Angriffsvektoren auf ein System, dem daraus resultierenden Umfang des erforderlichen Wissens und der somit notwendigen Spezialisierung der ausführenden Personen werden Penetrationstests meist im Team durchgeführt [10]. Hier findet manchmal auch der Begriff „Tiger Teams“ Verwendung; bei den Mitgliedern dieser Teams handelt es sich um sogenannte „White-Hats“, d.h. Sicherheitsexperten, die ihr Wissen weder zum eigenen Vorteil noch zum Schaden anderer nutzen.

Neben der Festlegung des anvisierten Zielsystems kann auch eine Kategorisierung des Tests vorgenommen werden. Grundsätzlich wird zwischen Black-Box- und White-Box-Ansatz unterschieden. Beim White-Box Ansatz sind dem Auftragnehmer detaillierte Informationen über das System zugänglich. Dies sind zum Beispiel Netzwerkpläne oder der Source Code einer zu testenden Applikation. Hingegen stellt der Black-Box Ansatz die Situation eines Außenstehenden nach [3]. In der Praxis hat sich als Mischform der sogenannte Grey-Box Ansatz sehr bewährt, da die Tester durch einige zusätzliche Informationen den Angriff präziser planen können und den Betrieb des Systems weniger gefährden.

Um die Testergebnisse vergleichbar und nachvollziehbar zu machen, wird beim Penetration Testing nach festgelegten Methoden, sogenannten „Vorgehensmodellen“, gearbeitet, die unter dem Punkt „Related Work“ näher beschrieben werden.

Sämtliche Modelle durchlaufen folgende Phasen:

- Aufklärung
- Scanning
- Zugang erlangen
- Dokumentation

Die empirische Analyse mehrerer Penetrationstests ergab, dass es sich entgegen der relativ starr wirkenden Vorgaben tatsächlich um sehr dynamische Workflows handelt. Da jeder Test individuell ist, lassen sich Phasen und Prozesse schwer im Vorhinein planen.

Unsere Arbeit behandelt daher folgende Punkte:

- Wir untersuchen und beschreiben die Vor- und Nachteile von kollaborativen Penetrationstests. Dabei legen wir besonderes Augenmerk auf die wachsende Professionalität in der „Cyberkriminalität“ und wie Sicherheitsexperten damit Schritt halten können.

- Wir analysieren die Anforderungen an ein Tool zur Unterstützung von kollaborativen Penetrationstests.
- Wir beschreiben unseren Prototypen, bei dem wir Teilaspekte bereits umgesetzt haben.

2. Problemstellung

Die Anzahl der gefundenen Schwachstellen eines Penetrationstests verhält sich in der Regel proportional zum Arbeitsaufwand, d.h. Qualifikation des Personals, Qualität der eingesetzten Tools und verfügbare Zeit. Durch Arbeitsteilung und Spezialisierung können die vorhandenen Ressourcen effektiv genutzt werden.

In der Vergangenheit hat sich gezeigt, dass die Angriffe auf Unternehmen an Professionalität gewinnen und sich diesbezüglich ein richtiger Markt entwickelt. Seinen Anfang hat dies bei „Vulnerability Research“ und der Entwicklung von Exploits. Die entdeckten Schwachstellen werden oftmals weiterverkauft bzw. versteigert wie zum Beispiel auf der WabiSabiLabi Plattform[3]. Eine weitere Sparte hat sich auf die Herstellung von Schadsoftware spezialisiert. Hierzu gehört die Entwicklung von Rootkit-Technologien zum Verstecken von Schadsoftware bzw. um Daten unbemerkt aus einem Unternehmensnetzwerk zu transportieren, so z.B. trojanische Pferde. Diese für kriminelle Zwecke entwickelte Software wird ständig weiterentwickelt. Mittlerweile ist Software-as-a-Service[6] in der Malware-Szene bereits umgesetzt.
Auf der nächsten Stufe steht der „Betreiber“, der diese Technologien einsetzt und so an für ihn interessante Daten (etwa Geschäftszahlen) gelangt oder die infizierten Rechner für Botnetze nutzt. Gestohlene Informationen können ebenso wie Ressourcen (siehe Botnetze) gesammelt und weiterverkauft werden [4].

2.1. Arten der Zusammenarbeit und Aufgabenverteilung

Die Komplexität heutiger Systeme und Netzwerke führt zu mannigfaltigen Angriffsvektoren, daher wird der Penetration-Test meist von mehreren Akteuren durchgeführt. Die Verteilung der Aufgabengebiete orientiert sich an folgenden Kriterien:

- funktional
- infrastrukturell
- prozessbezogen

Die funktionale Trennung erfolgt auf Basis der zu testenden Applikationen bzw. Systeme. Das verwendete Betriebssystem, Datenbanksoftware oder die eingesetzte Programmiersprache stellen hier die wesentlichen Parameter dar. Bei Tests, deren Schwerpunkt im Applikationsbereichs liegt, etwa Webapplikationen, ist die verwendete Programmiersprache der wichtigste Aspekt. Die Praxis zeigt, dass sich Security-Analysten, die Penetrationstests durchführen, zumeist auf fachliche Bereiche fokussieren. Die Schwerpunkte liegen hier meist auf Betriebssystem-, Applikations- oder Netzwerkebene. Eine funktionale Trennung bringt den Vorteil mit sich, dass für kritische Systeme hochspezialisierte Experten hinzugezogen werden können. So wäre ein denkbares Szenario: Im Laufe des Tests stellt sich heraus, dass es sich bei einem eingesetzten Mailsystem um Lotus Notes handelt. Ist nun ein Experte für Lotus Notes greifbar, kann dieser die Informationen rascher auswerten und die Penetration zielsicherer durchführen. Dies wiederum bedeutet eine Qualitätssteigerung der Ergebnisse und ein geringeres Ausfallsrisiko während der Durchführung des Tests.

[3] WabiSabiLabi Auktionsplattform: https://wslabi.com (Stand Juli 2008)

Eine infrastrukturelle Aufteilung erfolgt normalerweise anhand der IP-Adressen. Diese Trennung ermöglicht ein paralleles Arbeiten in jeder Phase des Tests. Besonders bei der Überprüfung von großen Netzwerken, bzw. ganzheitlichen Unternehmenspenetrationstests bedeutet diese Art der Arbeitsteilung einen Effizienzgewinn. Eine zusätzliche Segmentierung kann anhand einer Priorisierung der Kritikalität für das Unternehmen getroffen werden. Hochkrititsche Systeme sollten einerseits detailliert geprüft werden, andererseits muss dies oftmals auch in speziellen Zeitfenstern erfolgen.

Die prozessbezogene Betrachtung geht davon aus, dass gewisse Aktivitäten abgeschlossen sein müssen, damit mit weiteren Arbeitsschritten fortgefahren werden kann; ebenso gibt es Verzweigungen oder Entscheidungen, die auf den Ablaufpfad des Tests einwirken. Bei der Umsetzung von Vorgehensmodellen muss daher die Abhängigkeit der einzelnen Module analysiert werden, um dementsprechend die Ressourcen zu verteilen. Die Zuweisung der Tätigkeiten erfolgt für die Software-basierte Umsetzung idealerweise als Rollenkonzept, da sich dies gegenüber der benutzerorientierten Sicht als effizienter erwiesen hat.

Als vereinfachtes Beispiel für Abhängigkeiten von Aktivitäten sei der von außen durchgeführte Test eines internen Netzwerksegments genannt. Hierbei muss zuerst ein Scan des IP Bereichs durchgeführt werden, um herauszufinden, hinter welchen Adressen sich tatsächlich Systeme verbergen. In der Folge muss ein Gerät gefunden werden, das zumindest eine Schwachstelle aufweist, die ausgenutzt werden kann. Dieses muss in beiden Netzsegmente verbunden sein. Durch das Ausnützen der Schwachstelle kann logischer Zugriff auf das Gerät erlangt werden. Danach kann erst der interne Netzwerkbereich untersucht werden.

3. Related Work

Seit den ersten Penetrationstests durch Tiger Teams wurde kontinuierlich daran gearbeitet, die Reproduzierbarkeit des Vorgehens zu erhöhen und die Ergebnisse vergleichbar zu machen. Ebenso wurde versucht, Angriffsmöglichkeiten abzubilden. Attack Trees [16] stellen hierbei eine der bekanntesten Möglichkeiten dar. Daraus resultierend gibt es die Überlegung Attacken als Petri Netze darzustellen [11]. Bestehende Vorgehensmodelle dienen als Grundstein für unsere Überlegungen, da sie einerseits stets den Rahmenprozess mit den vergleichbaren Kernelementen wiedergeben, andererseits ebenso die verschiedenen Detailausprägungen dokumentieren.

Das NIST (National Institute of Standards and Technology) hat im Jahr 2003 „Guideline on Network Security Testing“ verfasst. Bei dieser Methode wird davon ausgegangen, dass die Informationsbeschaffung und der Angriff auf ein System oder Netzwerk iterative Prozesse darstellen, d.h. dass sich das Team während des Tests immer weiter vorarbeitet [13].

ISECOM (Institute for Security and Open Methodology) hat das OSSTMM (Open Source Security Testing Methodology Manual), aktuell in Version 2.2 frei verfügbar, veröffentlicht. Durch die ganzheitliche Betrachtung der IT-Sicherheit und Gliederung in einzelne Teilaspekte wird hier ein Rahmen für die funktionale Trennung vorgegeben [10].

Die OISSG (Open Information Systems Security Group) hat das ISSAF (Information Systems Security Assessment Framework) publiziert. Dieses Vorgehensmodell gliedert sich in 3 Phasen, wobei das tatsächliche Assessment in Phase 2 in detaillierte Arbeitspakete unterteilt wird. Die Dokumentation schließt auch die verwendeten Überprüfungstools mit ein [14].

Das BSI (Bundesamt für Informationstechnik) stellt mit dem „Durchführungskonzept für Penetrationstests“ eine Methode zur Abwicklung von Penetrationstests zur Verfügung. Das BSI misst der Informationsbewertung einen besonderen Stellenwert zu. In dieser Methode wird auch klar zwischen Modulen zur Informationsbeschaffung und jenen zum Eindringen unterschieden [3].

Die Fachgruppe Security der Schweizerischen Informatikgesellschaft (SI) hat im Jahr 1999 „Sicherheitsüberprüfungen von IT-Systemen mit Hilfe von ‚Tiger-Teams‘“ beleuchtet. Ein Tigerteam ist eine Gruppe von White - Hats Es handelt sich zwar um kein explizites Vorgehensmodell, allerdings werden zwei für unsere Arbeit relevante Punkte beleuchtet: die Durchführung im Team sowie die Unterteilung in Arbeitspakete [15].

4. Lösungsansatz

Das Ziel unserer Überlegungen und der Umsetzung durch unser Tool ist, durch Arbeitsteilung und damit einhergehende Spezialisierung eine Effizienzsteigerung und Qualitätsverbesserung zu erreichen. Um die Vorteile eines Collaborative Penetration Tests auch tatsächlich nützen zu können, minimieren wir die Reibungsverluste, die durch erhöhten Kommunikationsaufwand entstehen. Unser Prototyp kann als leichtgewichtiges, auf Penetration Tests spezialisiertes Groupware System beschrieben werden. Leichtgewichtig, da es ohne Installation sowie plattformunabhängig unter Windows, Linux und Unix Betriebssystemen betrieben werden kann. Es ist ausschließlich für Penetrationstests geeignet, da es aus Modulen besteht, welche genau auf diese Tätigkeit abgestimmt sind.

4.1. Vorgehensmodell und Prozess-Sicht

Die Verbindung zwischen Vorgehensmodell und Prozesssicht wird anhand eines Beispiels dargestellt. Abbildung 1 stellt einen Teilausschnitt, nämlich die Aufklärungsphase und partiell die Scanningphase eines vereinfachten Penetrationstests dar. Die einzelnen Subprozesse sind hierarchisch nummeriert. Im Subprozess 1.1.3 erfährt das Testteam, dass der Ansprechpartner für ein bestimmtes System das Unternehmen verlassen hat. Dies deutet auf einen möglichen Social-Engineering-Angriff hin. Es sollen so zusätzliche Informationen zur Kompromittierung des Systems erlangt werden. Der Subprozess 1.1.4 namens „Social Engineering“ wird initiiert. Hierfür sollte ein Spezialist für Social Engineering im Team eingesetzt und ihm diese Tätigkeit zugewiesen werden. Dies muss dem Team kommuniziert werden, anschließend sind diesem die Ergebnisse des Subprozesses zur Verfügung zu stellen.

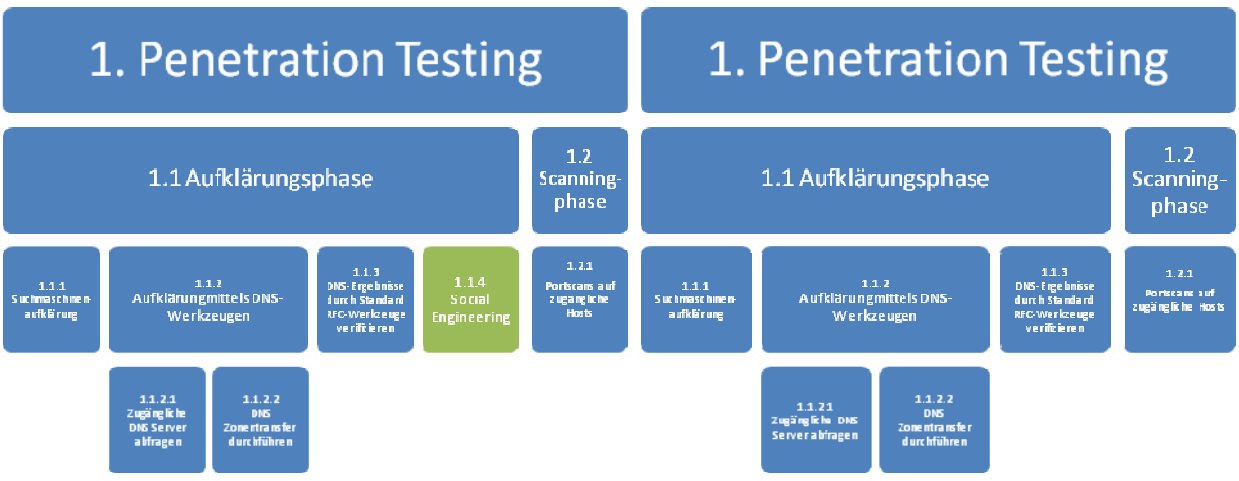

Abbildung 1: Beispielhafter Ablauf eines Penetrations-Tests (vgl. Winkler, 2008)

Die prozesstechnische Betrachtung von kollaborativen Penetrationstests hat folgende Aspekte aufgezeigt:

- Es existieren in der Regel sehr viele, kleine Prozessschritte;
- statisches Verhalten der Arbeitspakete auf Makroebene;
- sehr dynamisches Verhalten der Arbeitspakete auf Mikroebene;
- Kommunikation und Dokumentation sind Schlüsselpunkte.

Durch die feine Granularität der Prozessschritte ist es notwendig, diese in Arbeitspaketen zu aggregieren. Erst dadurch wird die zielführende Aufteilung auf die den Test durchführenden Security-Analysten ermöglicht.
Betrachtet man die Makroebene eines Penetrationstests, so gliedert sich dieser in Phasen, die je nach gewähltem Vorgehensmodell iterativ sein können. Diese Phasen stellen gebündelte Arbeitspakete dar. Da auch ein kollaborativer Penetrationstest an dieser Methode keine nennenswerten Änderungen verursacht, können diese in klassischen Prozessbeschreibungssprachen dokumentiert werden.
Auf Mikroebene erhöht sich bei der Bearbeitung durch mehrere Personen die Dynamik der Prozessaktivitäten wesentlich. Daraus resultieren Modifikationen, oftmals während der Durchführung. Es handelt sich dabei um sogenannte „ad hoc-Workflows“, die aus dem CSCW bekannt sind[1]. Diese weisen keine fixe Struktur auf, sondern können während der Durchführung von den handelnden Personen adaptiert werden [17].
Kommunikation ist bei einem kollaborativen Penetrationstest die wichtigste Komponente der Ressourcenallokation, da eine Detailplanung aufgrund der dynamischen Mikroebene nicht möglich ist. Da die Ressourcen bei diesen Tests durch das zur Verfügung stehenden Personal und den Zeitrahmen begrenzt sind, ist eine effiziente Nutzung zu gewährleisten.
Dokumentation ist die Basis für die Ergebnisaufbereitung. Sie muss transparent und vollständig sein. Da bei einem kollaborativen Penetrationstest mehrere Personen verteilt und teilweise parallel agieren, bestehen hier besonderen Herausforderungen[1]. Mit unserem Collaborative Penetration Testing Tool versuchen wir, dieses Problem zu lösen.

4.1.1. Kommunikationsunterstützung

Die Umsetzung von Erkenntnissen aus dem CSCW-Bereich ermöglicht es, die Reibungsverluste, die zuvor aufgezeigt wurden, zu kompensieren. Für die Realisierung unseres Konzepts ist sowohl eine synchrone als auch eine asynchrone Kommunikationsschiene notwendig. [7]
Mittels schriftlicher (synchroner) Mitteilungen können sämtliche Teammitglieder miteinander kommunizieren. In Kombination mit der notwendigen Dokumentation ergibt sich durch diese gleichzeitige Protokollierung des Informationsaustausches ein Nutzengewinn in Netzwerken verteilt arbeitender Personen[8].

Die während des Tests entstehenden Artefakte müssen auch (asynchron) zwischen den durchführenden Personen ausgetauscht werden können. Die Ergebnisse eines oder mehrerer Arbeitspakete stellen oftmals den Input für einen nächsten Arbeitsschritt dar, der unter Umständen von einer anderen Person ausgeführt wird. So kann etwa ein großflächiger Portscan den Input für gezieltes Fingerprinting darstellen.

Bezüglich Datenaustausch entschieden wir uns für den Einsatz von Peer-to-Peer-Technik (P2P). Diese bietet gegenüber einer klassischen Server-Client-Architektur mehrere Vorteile. Sämtliche im Test eingebundenen Hosts sollen äquivalente Partner sein. Durch den Einsatz von P2P erwarten wir

uns eine bessere Lastverteilung im Netzwerk. Während eines Tests befinden sich Hosts oftmals nicht im selben Netz, d.h. eine direkte Kommunikation ist nicht möglich. P2P ermöglicht es, andere Hosts als Brücken zu verwenden.

4.1.2. Koordinationsunterstützung

Ad hoc-Workflows erlauben die Definition und Adaption von Prozessen vor und während der Durchführung. Daher müssen folgende Funktionen unterstützt werden:

- Bearbeitung des Ablaufs vor und während des Tests
- Hinzufügen, Bearbeitung und Löschen von Arbeitsschritten vor und während des Tests
- Hinzufügen, Bearbeitung und Löschen von Arbeitspaketen vor und während des Tests

Hierfür werden Prozessvorlagen unterstützt, die über die P2P-Kommunikation zwischen den Teilnehmern ausgetauscht bzw. von diesen adaptiert werden können. Prozessvorlagen sind z.B. in XML verfasste Prozessbeschreibungen, die direkt in die Anwendung eingelesen und abgebildet werden können.

4.1.3. Compliance zu Penetration Testing Standards

Die Vorlagen, die für den Test erstellt werden, basieren auf den bekannten Vorgehensmodellen für Penetrationstests. Hier muss je nach Auftraggeber bzw. Testszenario das passende Modell ausgewählt werden.

4.1.4. Reporting

Das Reporting-Modul bereitet die Ergebnisse des Penetrationstests automatisiert auf. Wir haben uns für die Implementierung von zwei Arten von Bericht entschieden. In einem Management-Bericht werden die Daten aggregiert und aufbereitet. Die Darstellung erfolgt hier mittels Grafiken, um Entscheidungsträgern einen raschen Überblick zu vermitteln.
Der Detailreport dient als technischer Report für das Fachpersonal, damit dieses die Schwachstellen im Detail nachvollziehen und entsprechend gegensteuern kann.

4.1.5. Daten- und Kommunikationssicherheit

Bei der ausgetauschten Information bzw. den übermittelten Daten handelt es sich meist um streng vertrauliche Informationen. Da oftmals externe Tests stattfinden und die Kommunikation somit z.B. über das Internet abgewickelt wird, ist der Einsatz kryptographischer Methoden unbedingt notwendig.

4.1.6. Usability

Das Tool muss intuitiv bedienbar sein, um den Lernaufwand möglichst gering zu halten. Dies erhöht gleichzeitig die Benutzerakzeptanz.

4.1.7. Systemeinschränkungen

Unser Tool ist keine P2P-Tauschbörse. Es handelt sich um ein temporäres, d.h., während der Überprüfung existierendes Netz, das nur einem eingeschränkten Benutzerkreis, den Testern, zur Verfügung steht.
Die Applikation unterstützt nur einen kleinen Teil des Funktionsumfanges einer üblichen Workflow-Engine, außerdem ist kein aufwendiger Workflow-Editor implementiert. Im Vergleich zu einer Groupware-Plattform unterscheidet sich unser Tool durch die ausgedehnte Prozessunterstützung.

5. Design und Implementierung

Bei der Implementierung wurde durch ein modulbasiertes Design auf einfache Erweiterbarkeit geachtet. Wir ermöglichen somit auch die Einbindung von Fremdkomponenten, um besser auf sich ändernde Anforderungen eingehen zu können. Um die als Rich-Client-Applikation umgesetzte Lösung plattformunabhängig zu gestalten, wählten wir Python[4] als Programmiersprache. Sämtliche Parameter werden in Konfigurationsdateien ausgelagert, um die Wartbarkeit zu erleichtern.

5.1. Modulkonzept

Bei Penetrationstests wird oftmals eine beachtliche Anzahl von Tools eingesetzt, daher war es uns ein Anliegen, die Möglichkeit der Integration aufzuzeigen. Die Fremdkomponenten, wie zum Beispiel der Netzwerkscanner Nessus[5] oder der Portscann Nmap[6], wurden, durch das Einbinden in die Benutzeroberfläche durchgängig in den Workflow integriert. Die Module können dadurch Informationen wie z.B. Scan Ergebnisse oder Parameter austauschen, um den Ablauf des Prozesses zu verbessern.

5.2. Reporting

Das Reporting-Modul erzeugt auf Basis der Endberichte der einzelnen Module Berichte für verschiedene Zielgruppen. Das Ausgabeformat der Module ist unterschiedlich. Im Idealfall handelt es sich dabei um XML Dateien, die relativ einfach weiterverarbeitet werden können. Das Report-Modul parst die Ergebnisse und speichert diese akkumuliert und mit Metabeschreibungen in einer Datei ab. Aus dieser Metadatei wird je nach Berichtstemplate der spezifische Bericht für die jeweilige Zielgruppe erzeugt. Derzeit sind zwei Berichtsarten implementiert, eine Mangement Summary sowie ein technischer Report.

In der Management-Summary werden die aggregierten Daten für eine rasche Übersicht aufbereitet. Abbildung 2 zeigt eine sogenannte Vulnerability-Matrix. Hierbei werden mögliche Falsch-Positive aufgezeigt. Diese Auswertung basiert auf einem Vergleich der Ergebnisse zwischen Vulnerability-Scanner und Netzwerkscanner. Wir unterscheiden folgende zwei Fehlerarten:

- Typ 1 Fehler: Der Vulnerability-Scanner hat einen Host gefunden, der Netzwerkscanner jedoch nicht
- Typ 2 Fehler: Der Netzwerkscanner hat einen Host gefunden, der Vulnerability-Scanner jedoch nicht.

---

[4] Multiplattform Programmiersprache, die objekt-, aspektorientierte und funktionale Programmierung unterstützt.
[5] Nessus Vulnerability Scanner: http://www.nessus.org/nessus/ (Stand: Nov. 2008)
[6] Nmap Port Scanner: http://nmap.org/ (Stand: Nov. 2008)

Die Risiken der einzelnen Schwachstellen werden in einer dreistufigen Skala bewertet.

Vulnerability Matrix

	Vuln	Type 1	Type 2
High	5	0	
Medium	2	0	1
Low	28	0	

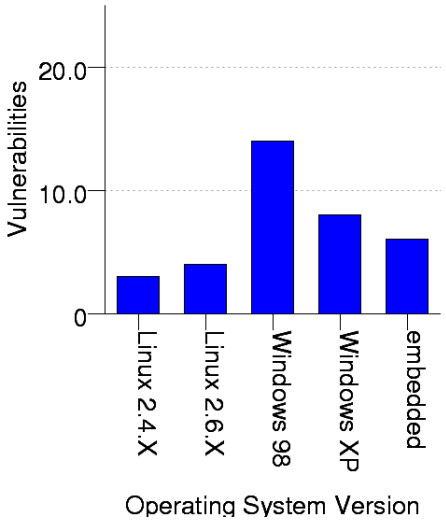

Abbildung 2: Management-Summary: Vulnerability Matrix und Schwachstellendiagramm

Die obige Abbildung zeigt die Anzahl der gefundenen Schwachstellen in Abhängigkeit des verwendeten Betriebssystems. Dies lässt z.B. Rückschlüsse auf den Reifegrad des Patchmanagements in den einzelnen Bereichen zu.

6. Zusammenfassung und Ausblick

In diesem Paper wurde Collaborative Penetration Testing, eine Weiterentwicklung der klassischen Penetrationstests nach Vorgehensmodellen, vorgestellt. Durch Arbeitsteilung und Spezialisierung wird die Effektivität und Effizienz erhöht, da die Ressourcen eines Penetrationstests durch das vorgegebenes Budget und die zur Verfügung stehende Arbeitszeit beschränkt sind. Dies stellt eine Möglichkeit dar, die wachsende Komplexität der zu testenden Infrastrukturen und der zunehmenden Professionalisierung im Cybercrime Bereich [5] [12] zu bewältigen.
Hierfür analysierten wir die grundlegende Problemstellung der Durchführung eines Penetrationstests aus prozesstechnischer Sicht. Durch die von uns angestrebte Arbeitsteilung ergibt sich Potential zur Minimierung kommunikationstechnischer Reibungsverluste. Mit unserem Prototyp stellen wir die praktische Umsetzung der Erkenntnisse dieses Bereichs im Kontext unserer Idee unter Beweis. Der Hauptvorteil ist ein qualitativ verbessertes Testergebnis.

Zur einfacheren und effizienteren Modellierung von Workflows ist der Einsatz der Business Process Execution Language (BPEL) [7] anzudenken. Dies würde die Definition der Vorgehensmodelle mittels einer standardisierten Beschreibungssprache ermöglichen.
Durch die Erstellung von Process Templates für mehrere Standards wäre es möglich, den Produktivitätsgewinn in Feldstudien empirisch zu belegen. Ebenso erwarten wir uns Rückschlüsse auf die anzustrebende Granularität im Prozessdesign, die wiederum relevante Aspekte für das zukünftige Design von Vorgehensmodelle für Penetrationstests liefern.

[7] BPEL 2.0 Spezifikation: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (Stand Juli 2008)

Weiters wird auch angedacht, die Art der Zusammenarbeit genauer zu analysieren[9], um die Unterstützung der Kommunikation weiter zu verbessern.

7. Literaturverzeichnis

[1] AVRAM, G., Of Deadlocks and Peopleware – Collaborative Work Practices in Global Software Development, International Conference on Global Software Engineering 2007

[2] BISHOP, M., FRINCKE, D., About Penetration Testing, IEEE Security & Privacy (Nov/Dec 07), 2007

[3] BSI, Durchführungskonzept für Penetrationstests, Bonn, 2003

[4] BSI, Lagebericht zur IT Sicherheit, http://www.bsi.de/literat/lagebericht/lagebericht2007.pdf (Stand Juli 2008), 2007

[5] DN1NJ4, RBN „Rizing“, http://www.shadowserver.org/wiki/uploads/Information/RBN_Rizing.pdf (Stand Juli 08), Shadowserver Foundation, 2008

[6] GRESCHLER, D., MANGAN T., Networking lessons in delivering “Software as a Service” Part 1, International Journal of Network Management Volume 12 Issue 5, 2002

[7] GROSS, T., KOCH, M., Computer-Supported Cooperative Work – Interaktive Medien zur Unterstützung von Teams und Communities, Oldenbourg Verlag, 2007

[8] HAYTHORNTHWAITE, C., Collaborative Work Networks among Distributed Learners, Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999

[9]HORROCKS; S., RAHMATI N., The Development and Use of A Framework for Categorising Acts of Collaborative Work, Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999

[10] HERZOG, P., Open-Source Security Testing Methodology Manual 2.2, ISECOM, http://www.isecom.info/mirror/osstmm.en.2.2.pdf (Stand Juli 2008), 2006

[11] MCDERMOTT, J.P., Attack Net Penetration Testing, Proceedings of the 2000 workshop on new security paradigms, Ballycotton, 2000

[12] MELANI, Lage in der Schweiz und international – Halbjahresbericht 2008/I , 2008

[13] NIST, Guideline on Network Security Testing, Washington, 2003

[14] OISSG, Information Systems Security Assessment Framework, 2004

[15] SCHWEIZER INFORMATIKGESELLSCHAFT, Sicherheitsprüfungen von IT-Systemen mit Hilfe von Tiger-Teams, 1999

[16] SCHNEIER, B., Attack Trees, Dr. Dobb’s Journal, 1999

[17] VAN DER AALST; W.; VOORHOEVE, M., Ad-Hoc Workflows: Problems and Solutions, Proceedings of the 8th International Workshop on Database and Expert Systems Applications, 1997

[18] WINKLER, S., Collaborative Penetration Testing, Universitätsbibliothek Universität Wien, Wien, 2008