HoneySens - Realisierung eines Honeypot-Netzes

Pascal Brückner
pascal.brueckner@tu-dresden.de
Thorsten Strufe
thorsten.strufe@tu-dresden.de

Zusammenfassung

Honeypots haben sich als probates Mittel zur Ergänzung bestehender IT-Sicherheitsinfrastrukturen etabliert. Ihre große Flexibilität erlaubt sowohl den Einsatz mittels öffentlicher IP-Adresse, aber auch den Betrieb innerhalb von Intranets, um Bedrohungen "aus dem Inneren" zu lokalisieren. Unter dem Projektnamen *HoneySens* wurde eine Honeypot-Sensorarchitektur für das Sächsische Verwaltungsnetzwerk konzipiert, implementiert und schließlich im praktischen Einsatz erprobt. Sie zeichnet sich dabei insbesondere durch ihre transparente Integration in bestehende IT-Landschaften aus. Erste Tests haben gezeigt, dass ein solches System neben tatsächlichen Angriffen auch auf Konfigurationsfehler im Netzwerk aufmerksam machen kann.

1 Einführung

Das Zeitalter der modernen Informationstechnik ist geprägt von einer sich ständig im Wandel befindlichen Bedrohungslage. Die Angriffe auf IT-Systeme sind dabei zumeist finanziell motiviert, weshalb insbesondere das Erzeugen wirtschaftlicher Schäden oder der Abfluss von verwertbaren Informationen aus Sicht von Eindringlingen im Mittelpunkt stehen. Mit dem Anwachsen der Internetkriminalität zu einem eigenständigen Geschäftsfeld und dem damit verbundenen Wildwuchs an Malware sind herkömmliche, zumeist signaturbasierte Schutzsysteme wie Anti-Viren-Software oder Intrusion-Detection-Systeme überfordert. Honeypots als passive, genauestens überwachte und bewusst verwundbare

Netzwerkressourcen können bestehende Schutzsysteme ergänzen und wertvolle Informationen über verdächtige Aktivitäten innerhalb von Netzwerken sammeln, die ansonsten verborgen bleiben.

Das Sächsische Verwaltungsnetzwerk (*SVN*) ist ein leistungsfähiges, weiträumiges Kommunikationsnetz für den Informationsaustausch innerhalb der sächsischen Landesverwaltung. Sämtliche angeschlossene Ressorts und Behörden sind dabei in Form von logischen, eigenständigen Netzsegmenten vertreten. Das SVN stellt allen Teilnehmern eine große Zahl von gemeinsam genutzten Informations- und Kommunikationsdiensten zur Verfügung, aber auch zwischen den verschiedenen Teilnetzen existieren zahlreiche, individuell genutzte Netzübergänge. Die resultierende komplexe Netzarchitektur wird dezentral administriert, so dass jedes SVN-Mitglied auch für die Absicherung des eigenen Subnetzes verantwortlich ist. In diesem Rahmen wurden bereits herkömmliche IT-Sicherheitslösungen wie Firewalls in großem und Intrusion-Detecion-Systeme (*IDS*) in begrenztem Umfang implementiert. Letztere benötigen für ihren Betrieb eine Vielzahl an Ressourcen, sowohl monetär als auch personell, weshalb ein Einsatz solcher Lösungen nur an zentralen Knotenpunkten mit hohem Datenaufkommen in Erwägung gezogen wird. Eine vollständige Absicherung kleinerer Teilbereiche, beispielsweise eines gesonderten Büronetzes, gestaltet sich unter diesen Bedingungen aber als schwierig. Aus diesem Grund wurde vom Beauftragten für Informationssicherheit des Landes der Einsatz von Honeypots in Erwägung gezogen, um einen zusätzlichen Blickwinkel auf einzelne Teilnetze zu erhalten.

Ziel der Arbeit war die Konzeption und Implementierung einer auf Honeypots basierenden Monitoring-Lösung für den Einsatz innerhalb von komplexen IT-Landschaften. Honeypots werden definiert als genauestens überwachte Netzwerkressourcen, die von potentiellen Angreifern analysiert, angegriffen und auch kompromittiert werden sollen [4]. Durch diese direkte Interaktion mit potentiellen Eindringlingen ist es mit solch Instrumenten möglich, vielfältige Informationen über Angriffe zu gewinnen, die über herkömmliche IDS nicht zur Verfügung stünden.

Honeypots sind außerdem typischerweise rein passive Netzwerkteilnehmer ohne eine globale Sicht auf den Datenverkehr im Netzwerk. Folglich kann allein der Versuch, mit Honeypots zu kommunizieren, bereits als Angriffsversuch gewertet werden. Diese zusätzliche Sicht auf Netzwerke sollte möglichst effizient und kostengünstig für den großflächigen Einsatz innerhalb von IT-Landschaften wie dem SVN umgesetzt werden. Im Kontrast zu Honeypots, die traditionell

besonders zur Detektion von Angriffen aus dem Internet Verwendung finden, sollte das System dahingehend konzipiert werden, Angriffe aus dem Inneren eines Netzwerkes heraus zu registrieren. Es ist somit fähig, auf sich selbstständig ausbreitende Malware aufmerksam zu werden, die beispielsweise über verseuchte USB-Sticks eingeschleust wurde. Aber auch manuelle Einbrüche oder Aktivitäten von Innentätern können mit derartigen Maßnahmen sichtbar werden.

Im Rahmen eines Forschungsprojektes in Kooperation mit dem Sächsischen Ministerium des Innern (SMI) wurde ein Prototyp für ein Honeypot-Netzwerk entwickelt, das insbesondere den Anforderungen des Verwaltungsnetzwerkes gerecht werden sollte. So stand die möglichst transparente Integration der Lösung in das bestehende Netzwerk im Vordergrund, aber auch die einfache Wartbarkeit und Benutzerfreundlichkeit waren wesentliche Kernforderungen.

2 Verwandte Arbeiten

Die Infrastruktur des SVN erschwert insofern den Einsatz von Honeypots, als dass für diese kein separates Subnetz (“Honey-Netz”) vorgesehen ist, das sich ausschließlich aus Honeypot-Knoten zusammensetzt und in das fragwürdiger Datenverkehr zur Analyse weitergeleitet werden kann. Dieser Ansatz erlangte insbesondere durch das von Nils Provos entwickelte Honeypot-Framework **honeyd** [3] Verbreitung, das die Realisierung eines solch abgeschlossenen Netzes wahlweise physischer oder virtueller Natur im Vergleich zu früheren Tools deutlich erleichtert hat.

Einen alternativen Ansatz stellen die Forschungsprojekte **Collapsar** [2] und **Potemkin** [5] vor, bei denen sogenannte *Redirector* in verschiedenen Netzwerken platziert werden und eingehenden Datenverkehr über VPN- oder GRE-Tunnel an eine zentrale Instanz weiterleiten (*Collapsar Frontend*, bzw. *Potemkin Gateway*). Dort werden bei Bedarf virtuelle Honeypots erzeugt, um die Anfragen zu verarbeiten und deren Antworten wieder über den Tunnel zurück ins Ursprungsnetz geleitet. Management und Auswertung der gesammelten Daten erfolgt ebenfalls an diesem zentralen Knoten. Mit einer solchen Architektur wird das klassische Prinzip der Honey-Netze aufgebrochen: eine große Anzahl von Honeypots kann effizient in vielen Zielnetzen gleichzeitig betrieben werden. Nachteilig ist bei diesem Verfahren jedoch, dass Angreifer aufgrund der höheren Latenzen die Honeypot-Knoten identifizieren können und eine große Zahl von Tunneln permanent aufrecht erhalten werden muss.

Sensor	Pakete	ICMP	TCP
bb1	9	9	0
bb2	358	358	0
bb3	28	10	18
bb4	57	45	12
bb5	43	43	0
bb6	347	288	59

Tabelle 1: Akkumulierte Sensordaten

In deutschen CERT-Verbund existiert seit 2007 außerdem das Gemeinschaftsprojekt **CarmentiS** [1], in dessen Rahmen Sensoren u.a. in internen Netzwerken von Unternehmen platziert werden, die ebenfalls nach dem Honeypot-Prinzip arbeiten und Informationen über Verbindungsversuche an einen sog. *Konzentrator* weiterleiten, der diese Daten für eine allen Teilnehmern zugängliche *Frühwarnzentrale* aufbereitet.

3 Konzeption

Zunächst galt es, für das zu entwerfende Honeypot-Netz ein Modell eines Angreifers zu entwickeln, der mit Hilfe der Monitoring-Lösung detektiert werden soll. Im Fokus lagen dabei die Detektion von sich lokal selbstständig ausbreitender Malware, aber auch die Aktionen von Innentätern und menschlichen Eindringlingen sollten prinzipiell erkennbar sein. Es war somit grundsätzlich kein allmächtiger Angreifer vorgesehen. Er kann aber sehr wohl einzelne Knoten im Netzwerk übernehmen und von diesen auch Angriffe starten. Es wird allerdings davon ausgegangen, dass ein solcher Eindringling zwar vollen Netzwerkzugriff durch die von ihm übernommenen Systeme besitzt, aber aufgrund der typischerweise hierarchischen Netzwerkstruktur nicht in allen Netzen zugleich präsent sein kann. Es galt zusätzlich die Einschränkung, dass Angreifer keinen physischen Zugriff auf die Honeypots besitzen.

Vor dem Entwurf der Architektur wurde zunächst eine kurze praktische Analyse der Menge des zu erwartenden Netzwerkverkehrs in verschiedenen Teilnetzen des SVN ausgeführt. An den Messpunkten wurden Rechner für eine Dauer von sieben Wochen installiert, die alle per Unicast an sie gerichteten Pakete aufzeichneten. Die aus Gründen des Datenschutzes abstrahierten Ergebnisse sind in Tabelle 1 aufgeführt.

Zusammengefasst stellte sich heraus, dass das tatsächliche Datenaufkommen unter den gegebenen Bedingungen verhältnismäßig gering ausfällt. Die Sensoren 2 und 6 mit dem erhöhten Datenaufkommen waren zudem auch über öffentliche IP-Adressen erreichbar, was die größeren Paketmengen begründet. Die Anforderungen an die von den zukünftigen Honeypots zur Verfügung gestellten Ressourcen viel folglich eher gering aus. Weiterhin waren Eigenheiten des SVN zu berücksichtigen: das weiträumige Netzwerk besteht aus unzähligen dezentral verwalteten Teilnetzen, einer Reihe zentraler Servernetze und einem gemeinsamen Internetzugang. Alle Netzübergänge sind mit restriktiven Firewalls besetzt, die bis auf gesonderte Ausnahmen keinen eingehenden Datenverkehr zulassen. Zusätzlich steht der Zugang ins Internet ausschließlich über netzspezifische Proxy-Server zur Verfügung. Eine Kernforderung beim Entwurf des Systems war, dass die bestehende Netzinfrastruktur möglichst nicht für den Einsatz der Honeypots umkonfiguriert werden muss. Die Integration des Systems sollte daher vorzugsweise transparent während des Produktivbetriebes erfolgen.

Der resultierende Architekturentwurf (siehe Abbildung 1) orientiert sich an den in Kapitel 2 vorgestellten Systemen Potemkin und Collapsar und sieht eine Reihe von sog. *Sensoren* vor, die innerhalb aller zu überwachenden Netzwerke platziert werden und verdächtige Ereignisse unter Berücksichtigung der Firewall-Richtlinien an einen zentralen, von allen Punkten aus erreichbaren Server melden. Da die Sensoren aufgrund der Firewalls nicht von außerhalb ihrer Teilnetze kontaktiert werden konnten und eine Rekonfiguration der Regelsätze zu vermeiden war, ist es Aufgabe der Sensoren, die Verbindung mit dem Server zu initiieren. Dies umfasst sowohl das Melden aufgezeichneter Vorfälle, aber auch ein periodiosches Polling, um Konfigurationsänderungen in Erfahrung zu bringen.

Bei der Kommunikation zwischen Sensoren und Server wird der Fakt ausgenutzt, dass es die Firewalls zulassen, dass von allen Teilnetzen aus HTTP(S)-Verbindungen über die Proxy-Server zu den zentralen Serverdiensten und ins Internet möglich sind: das Serversystem stellt eine HTTPS-Schnitstelle bereit, die von allen Sensoren aus kontaktiert werden kann. Um die Sensor-Architektur weitestgehend unabhängig von sonstigen Netzwerkkomponenten zu halten, werden organisatorische Aktivitäten wie beispielsweise die Zeitsynchronisation der Geräte ebenfalls über den gesicherten Kanal zum Server abgewickelt. Hier hilft der Umstand, dass Web-Server mit jeder Anfrage auch einen aktuellen Zeitstempel mitliefern. Durch die resultierende Latenz auftretende geringe

Abweichungen von der Serverzeit stellten sich beim Testbetrieb als unproblematisch heraus.

Am serverseitigen Endpunkt ist eine REST-API vorgesehen, über die sowohl Sensoren als auch Administratoren entsprechend ihrer Berechtigungen mit dem System interagieren können. Zusätzlich stellt der Server eine webbasierte Benutzerschnittstelle zur Verfügung, die wiederum auf der REST-API aufsetzt und eine zeitgemäße und benutzerfreundliche Interaktion mit der Software ermöglicht. Dies umfasst sowohl die Verwaltung als auch die Auswertung der aufgezeichneten Daten.

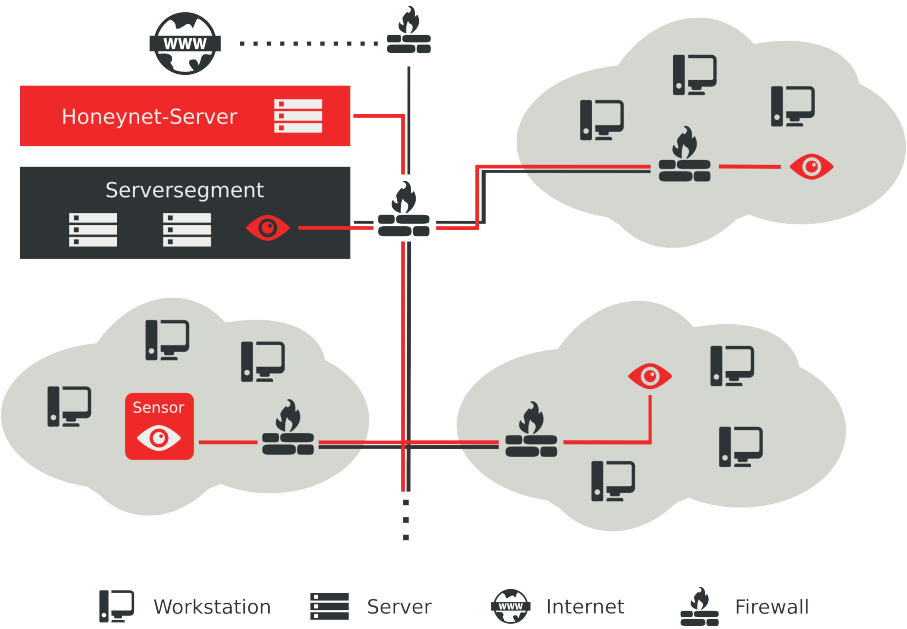

Abbildung 1: Architektur

Hauptaufgabe des Servers ist die Verwaltung aller registrieren Sensoren, was auch vollständige Aktualisierungen der Sensorsoftware, sowie die Konfiguration der angebotenen Honeypot-Dienste umfasst. Erfasste Ereignisse sind dem Administrator übersichtlich zu präsentieren und statistisch zusammenzufassen. Häufig wiederkehrende, aber als harmlos klassifizierte Ereignisse können mit Hilfe spezieller Filterregeln bewusst ignoriert werden. Funktionen zur Mandantenfähigkeit ermöglichen zudem die Aufteilung der Benutzer, Sensoren und registrierten Ereignisse in Gruppen und ermöglichen es, eine einzelne

Server-Instanz für eine Vielzahl von administrativ voneinander unabhängigen Domänen zu nutzen.

Da die Verwaltung des Sensornetzes und auch die Auswertung der gesammelten Daten an einem zentralen Punkt erfolgt, galt es, eine Reihe von Schutzzielen für die resultierende Architektur selbst zu definieren: Die Integrität und Vertraulichkeit der Datenübertragung zwischen Sensoren und Server muss gewährleistet sein, um zu verhindern, dass Angreifer die übertragenen Ereignisdaten beeinflussen können. Weiterhin ist es essentiell, die Integrität der Ausgaben des Servers zu gewährleisten und auch die Integrität der Daten, die für einen bestimmten Nutzer des Systems im Rahmen der Mehrbenutzerfähigkeit *nicht* zugänglich sind, sicherzustellen. Es wird hier ein Angreifer mit vollem Netzwerkzugriff angenommen, der keinen physischen Zugriff auf die Systeme besitzt. Das Wartungs- und Administrationspersonal der Komponenten wird zudem als vertrauenswürdig angenommen und kann nur im Rahmen seiner zugewiesenen Domänen handeln.

4 Umsetzung

Die Realisierung der vorgestellten Architektur erfolgte serverseitig mit Hilfe gängiger Web-Technologien. Die sowohl von den Sensoren als auch einer Web-Anwendung konsumierte RESTful API ist eine klassische CRUD-Anwendung, in der Administratoren über Zugangsdaten und Session-Cookies authentifiziert werden, während für die Sensoren eine gesonderte PKI die Authentizität der empfangenen Nachrichten sicherstellt. Das System ist weiterhin mandantenfähig, so dass jeder Nutzer nur auf Ressourcen seiner eigenen Domäne(n) zugreifen kann. Die Kommunikation zwischen Sensor und Server ist zudem mit TLS abgesichert, wobei seitens der Sensoren zusätzlich Certificate Pinning betrieben wird, um die Authentizität des Server-Endpunktes zu gewährleisten.

Alle Sensoren kontaktieren den Server in einem frei definierbaren Intervall, um ihre Konfiguration und Softwarestand mit den Nutzervorgaben zu synchronisieren. Neben der Verwaltungssoftware für den Betrieb der Sensoren bieten diese außerdem eine Reihe von Honeypot-Diensten an, die für das eigentliche Sammeln der kritischen Daten verantwortlich sind. Dabei handelt es sich ausschließlich um Low-Interaction-Honeypots, also Simulationen verschiedener Serverdienste, deren Einsatz mit einem geringerem Gefahrenpotential und Ressourcenbedarf im Vergleich zu High-Interaction-Honeypots verbunden ist:

recon : Ein im Rahmen des Projektes entwickelter Dienst, der für alle nicht durch andere Services belegte TCP-Ports Verbindungen annimmt und nach dem ersten empfangenen Datenpaket wieder schließt. UDP-Pakete werden lediglich aufgezeichnet, aber nicht beantwortet. Alle auf diese Art gewonnenen Informationen inklusive der Paketinhalte werden nach erfolgter Kommunikation an den Server weitergeleitet. Zum Schutz vor DoS-Angriffen besitzt der recon-Dienst außerdem rudimentäre Erkennungsroutinen für Scanversuche, so dass einzelne Quellhosts bewusst temporär ignoriert werden, wenn diese zu viel Datenverkehr auf dem Sensor erzeugen.

cowrie : Ein Open-Source-Fork des SSH-Honeypots *kippo*, der das SSH-Protokoll simuliert und Angreifern auch das Anmelden auf dem Honeypot und das Ausführen einer virtuellen, stark limitierten Shell ermöglicht. Das Projekt wurde um Module zur Kommunikation mit dem HoneySens-Server erweitert.[1]

dionaea : Ein weiterer Open-Source-Honeypot mit Unterstützung für eine Vielzahl häufig verwendeter Protokolle. Wird im Rahmen dieses Projektes ausschließlich für die Simulation von SMB/CIFS-Diensten genutzt, wie sie typischerweise in Windows-Netzwerken anzutreffen sind. Das Projekt wurde ebenfalls um ein Modul für die Kommunikation mit dem HoneySens-Server erweitert.[2]

Da der Austausch zwischen Server und Sensoren via HTTPS stattfindet, wurde als Datenformat die von Webanwendungen häufig genutzte *JavaScript Object Notation* (JSON) gewählt. Es kommt für alle Nachrichtentypen (Ereignisse, Status-, Konfigurationsdaten) ein individuelles Nachtrichtenschema zum Einsatz, mit dem auch mehrere Ereignisse, die in kurzer Zeit an einem Sensor registriert wurden, gemeinsam übertragen werden können. Dabei werden niemals Pakete einfach vollständig weitergeleitet, sondern vom Sensor zuvor ausgewertet und nur potentiell relevante Komponenten, wie beispielsweise die UDP- oder TCP-Payload, unverändert an den Server übertragen.

Die Hardwareplattform für die Sensoren ist prinzipiell flexibel, im Rahmen des Forschungsprojektes wurden aber ausschließlich mit dem *BeagleBone Black*[3] kompatible Einplatinen-Computer genutzt. Diese Systeme besitzen

[1] `https://github.com/micheloosterhof/cowrie`
[2] `https://github.com/rep/dionaea`
[3] `https://beagleboard.org/black`

zusätzlich zur externen SD-Karte noch einen internen Flash-Speicher, was für die Umsetzung atomarer System-Aktualisierungen genutzt wird, die zentral vom Server aus für alle Sensoren angestoßen werden können.

Das Deployment des Serversystems wird mit Hilfe von containerbasierter Virtualisierung realisiert: Die Software steht in Form eines *Docker*-Images zur Verfügung, das in wenigen Schritten zum Erstellen eines neuen Containers genutzt werden kann. Assistenzprozesse unterstützen den Administrator zudem bei der Installation und Aktualisierung der Serversoftware. Um die Integrität der Serverausgaben sicherzustellen, sind zusätzlich von vertrauenswürdigem Personal durchgeführte Härtungsmaßnahmen des Hostsystems Bestandteil der Installationsroutine.

Nutzer können aufgetretene Vorfälle mit Hilfe der Webanwendung auswerten, aber auch automatisiert in regelmäßigen Abständen oder bei kritischen Ereignissen (wenn einer der Honeypot-Dienste direkt kontaktiert wurde) eine Benachrichtigung per E-Mail erhalten.

5 Evaluation

Mit Hilfe der vorgestellten Architektur ist es möglich, jeglichen direkt an die mit einer IP-Adresse versehenen Sensoren gerichteten IP-Datenverkehr aufzuzeichnen. Broadcasts und Multicasts werden im Sinne der Anforderungen und nach Abschätzung des Bedrohungspotentials ignoriert. Zusätzlich erlauben die auf den Sensoren betriebenen Honeypot-Dienste, genauere Informationen über auffällige SSH- und SMB-Aktivitäten zu gewinnen. Für die Evaluation wurde das System nach Sicherstellung der grundsätzlichen Funktionalität mit rudimentären Tests (Verbindungsaufbau mit netcat, Angriffe über metasploit) und im Anschluss auch direkt innerhalb produktiver Netze eingesetzt, um unmittelbar praktische Erfahrungen sammeln zu können.

Der Probeeinsatz erstreckt sich seit Anfang 2016 über das Universitätsnetz der TU Dresden und mehrere Teilnetze des SVN. Im TU-Netz wurden bis zum Jahresende 15 Sensoren installiert, während im Verwaltungsnetz im gleichen Zeitraum etwa 8 Sensoren installiert wurden. Das *HoneySens* getaufte System konnte seine Eignung zur Erkennung verdächtiger Vorfälle erfolgreich unter Beweis stellen, da die jeweiligen Administratoren mit dessen Hilfe auf eine Reihe von sicherheitskritischen Vorfällen aufmerksam wurden. Während einzelne Vorfälle an dieser Stelle unter Berücksichtigung des Datenschutzes nicht im Detail diskutiert werden können, waren unter den interessanten Er-

eignissen HTTP-Requests auf untypischen Ports (weder TCP/80, TCP/8080 noch TCP/443), diverse Portscans und auch Zugriffe auf Port TCP/445 (SMB) enthalten. Eine exakte Bewertung der einzelnen Vorfälle ist im Rahmen dieser Produktivnetze schwierig, da eine Untersuchung der verdächtigen Systeme aus Datenschutzgründen meist nicht ohne weiteres möglich war.

Neben dieser offensichtlichen Eignung als Frühwarnsystem hat sich aber zudem herausgestellt, dass mit Hilfe des Sensorsystems insbesondere auch Konfigurationsfehler im Netzwerk sichtbar werden. So ermöglichten beispielsweise eine Reihe von Routen, die vor langer Zeit als Ausnahmeregeln durch einige Firewalls hindurch zugelassen und dann vergessen wurden, die Kommunikation zwischen zwei eigentlich strikt zu trennenden Produktivnetzen. Da dieser Weg auch von einigen Hosts genutzt wurde und letztendlich auch der Sensor kontaktiert wurde, konnten diese Probleme leicht ausfindig gemacht werden.

Insgesamt hat sich herausgestellt, dass nach dem Einsatz neuer Sensoren zunächst eine begrenzte “Anlernphase” erforderlich ist, in der das System aktiv beobachtet und bei Bedarf häufig wiederkehrende Ereignisse in die Whitelist eingepflegt werden müssen. Unterlässt man diesen Schritt, ist mit einer großen Zahl von False-Positives zu rechnen. So kontaktieren beispielsweise viele Drucker aus bislang unbekannten Gründen die Sensoren regelmäßig auf Port UDP/137 mit ansonsten leeren Paketen. Außerdem waren in einigen Teilnetzen anderweitige der IT-Sicherheit dienliche Systeme installiert, die regelmäßig alle verfügbaren Hosts enumerierten. Derartige TCP- oder UDP-Scans wurden als Ereignisse registriert und konnten von den Administratoren als harmlos klassifiziert und in Zukunft ignoriert werden.

Bezüglich der Skalierbarkeit des Systems traten in den bisherigen Ausbaustufen keinerlei Probleme auf. Die REST-API und somit auch das Webinterface nutzen an allen wichtigen Stellen Lazy Loading, um die zu Clients übertragenen Datenmengen zu begrenzen. Es waren zudem für den praktischen Einsatz eine Reihe von bürokratischen Hürden zu überwinden und Vorurteile gegenüber einem derartigen Sensorsystem auszuräumen. Transparenz hatte dabei höchste Priorität: Manche Administratoren haben den Zweck des Systems missverstanden und Honeypots als eine Überwachungsmaßnahme ihrer Arbeitgeber interpretiert. Hochwertige Dokumentation und Schulungssitzungen der zukünftigen Nutzer sind folglich nicht zu unterschätzen.

6 Zusammenfassung

Im Rahmen eines Forschungsprojektes in Kooperation mit dem Sächsischen Ministerium des Inneren ist das Honeypot-Sensornetzwerk namens *HoneySens* entstanden, das als Frühwarnsystem zusätzlich zu bestehenden Sicherheitslösungen wie Firewalls und IDS eine zusätzliche Sicht auf das Netzwerk geben kann. Der Fokus liegt auf der Detektion von verdächtigen Zugriffen durch Angreifern, die aus dem Inneren eines Netzwerks operieren, also zentrale Firewalls bereits überwunden haben.

Für den Entwurf waren verwandte Arbeiten zu verteilten Honeypot-Netzen Vorbild, wobei zusätzlich auch die besonderen Beschränkungen für Kommunikationsmöglichkeiten innerhalb von komplexen IT-Landschaften wie dem Sächsischen Verwaltungsnetzwerk berücksichtigt wurden. Beim praktischen Betrieb hat sich herausgestellt, dass nach einer initialen Anlernphase, bei der eine Whitelist zur Vermeidung von False-Positives gepflegt werden muss, das System nur noch wenige, dann aber auch tatsächlich verdächtige Ereignisse aufzeichnet. Zusätzlich wurde deutlich, dass mit Hilfe der Sensoren auch Konfigurationsfehler, die nicht direkt auf Schadsoftware oder anderweitige Angreifer zurückzuführen sind, ans Tageslicht kommen können.

Zukünftiger Forschungsschwerpunkt ist die Verbesserung der Sichtbarkeit einzelner Honeypot-Sensoren, um die Wahrscheinlichkeit eines aufgezeichneten Kontaktversuchs durch Angreifer gezielt zu erhöhen. Hierfür sollen verschiedene Ansätze, wie beispielsweise die dynamische Ausweitung des abgedeckten Adressbereiches durch eine Abwandlung des ARP-Spoofings untersucht werden.

Literatur

[1] Bernd Grobauer, Jens Ingo Mehlau, and Jürgen Sander. Carmentis: A co-operative approach towards situation awareness and early warning for the internet. In *IMF*, pages 55–66, 2006.

[2] Xuxian Jiang, Dongyan Xu, and Yi-Min Wang. Collapsar: a VM-based honeyfarm and reverse honeyfarm architecture for network attack capture and detention. *Journal of Parallel and Distributed Computing*, 66(9):1165–1180, 2006.

[3] Niels Provos and Thorsten Holz. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional, first edition, 2007.

[4] L. Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[5] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C Snoeren, Geoffrey M Voelker, and Stefan Savage. Scalability, fidelity, and containment in the potemkin virtual honeyfarm. *ACM SIGOPS Operating Systems Review*, 39(5):148–162, 2005.