Anwendung von maschinellem Lernen zum automatischen Erkennen von Padding-Orakel-Seitenkanälen

Dr. Claudia Priesterjahn
achelos GmbH, Vattmannstraße 1, 33100 Paderborn, claudia.priesterjahn@achelos.de
Jan Peter Drees
Bergische Universität Wuppertal, Gaußstraße 20, 42119 Wuppertal, jan.drees@uni-wuppertal.de
Pritha Gupta
Universität Paderborn | SICP, Zukunftsmeile 2, 33102 Paderborn, prithag@uni-paderborn.de
Dr. Simon Oberthür
Universität Paderborn | SICP, Zukunftsmeile 2, 33102 Paderborn, oberthuer@sicp.de

1 Motivation

Vernetzte Geräte finden sich ganz selbstverständlich in fast allen Lebenslagen und ihre Anzahl wächst immer noch stetig: Seien es Smartphones, Geräte im Smart Home oder die massive Vernetzung von Industrieanlagen und kritischen Infrastrukturen. Ein Großteil dieser Geräte enthält Informationen, die schützenswert sind oder sie sind Bestandteil von Anlagen, deren Ausfall oder Kompromittierung durch einen Cyberangriff schwerwiegende Folgen bis hin zum Verlust von Menschenleben haben kann.

Aus diesem Grund ist eine sichere Kommunikation zwischen diesen vernetzten Geräten unerlässlich. Ein Eckpfeiler der sicheren Kommunikation im Internet sind kryptographische Protokolle. Sie ermöglichen es, die Vertraulichkeit, Integrität und Authentizität von übertragenen Daten zu schützen. Ein kryptographisches Protokoll bietet allerdings nur dann die erwartete Sicherheit, wenn es vollständig korrekt implementiert ist. Selbst für sehr gut untersuchte und sehr gut verstandene Protokolle, wie zum Beispiel TLS, werden seit vielen Jahren immer wieder neue Schwachstellen und Seitenkanäle entdeckt. Zudem kann jede Änderung der Implementierung neue Seitenkanäle öffnen. Ein kontinuierlich wirksamer Schutz der IT-Sicherheit erlangt somit in Unternehmen eine immer höhere Notwendigkeit.

In der Computersicherheit ist ein Seitenkanalangriff ein Angriff, der auf Informationen beruht, die aus der Implementierung des Computersystems gewonnen wurden, und nicht auf einer direkten Schwäche im implementierten Algorithmus selbst. Diese Seitenkanäle zuverlässig zu erkennen ist eine offene Herausforderung.

Eine bestehende Möglichkeit der Erkennung von Seitenkanälen ist die manuelle Untersuchung durch IT-Sicherheits-Experten. Angesichts der großen Verbreitung von kryptographischen Protokollen wie TLS, welches heutzutage in der Form von HTTPS auf praktisch jedem smarten Gerät benutzt wird, ist dies keine dauerhafte Lösung: Da sich bei jedem Software-Update ein neuer Fehler einschleichen kann, ist kontinuierliches Testen nötig.

Unser Beitrag umfasst die Vorstellung eines Verfahrens, das wir gemeinsam im Verbundprojekt AutoSCA[1] entwickelt haben, sowie dessen prototypische Integration in ein Produkt. Durch die Kombination von automatisiertem Testen und maschinellem Lernen ist unser Verfahren in der Lage, Seitenkanäle in TLS-Software zu detektieren, um so künftig die oben genannten schwerwiegenden Sicherheitslücken zu vermeiden [1]. Dabei analysieren wir, ob anhand des beobachtbaren verschlüsselten Netzwerkverkehrs Rückschlüsse auf die verschlüsselten Daten gezogen werden können. Dies würde auf einen Seitenkanal und damit auf eine potenziell schwerwiegende Sicherheitslücke hindeuten.

Unser Verfahren kann für sämtliche TLS-Software eingesetzt werden, unabhängig von der Art der Implementierung, der Hardware oder dem Betriebssystem, da die Tests und maschinellen Lernverfahren auf Protokollebene ausgeführt werden. Durch die Integration des entwickelten Verfahrens in den TLS Inspector der achelos[2] wird das Verfahren in die breite Anwendung gebracht. Zudem stellen wir unsere gesamte Implementierung[3] kostenlos und quelloffen zur Verfügung. Dazu gehört auch, dass Teile des ehemals kostenpflichtigen, nicht quelloffenen TLS Test Tools[4] der achelos kostenlos und quelloffen veröffentlicht werden. Damit möglichen wir einen leichten Zugriff für Wissenschaftler und Open Source Entwickler.

2 Verfahren

Im ersten Teil unseres Projekts haben wir uns mit einer speziellen Art von Seitenkanalangriffen beschäftigt: den Padding-Orakel-Angriffen. Mit unserem Verfahren kann automatisch und mit hoher Sicherheit vorhersagt werden, ob ein TLS-Server durch bisher bekannte und auch unbekannte Padding-Orakel-Angriffe angreifbar ist. Der Padding-Orakel-Angriff ist ein Man-in-the-Middle-Angriff, wobei der

[1] https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/autosca
[2] https://www.achelos.de
[3] https://github.com/ITSC-Group/autosca-tool
[4] https://www.achelos.de/de/tls-test-tool.html

Angreifer aus dem Padding[5] einer mit PKCS#1v1.5-gepaddetem RSA verschlüsselten Nachricht Informationen erhalten kann, die zur Entschlüsselung der Nachricht genutzt werden können. Der Angreifer manipuliert gezielt das Padding abgefangener verschlüsselter Nachrichten und sendet diese an den Server. Der Server meldet entweder eine fehlerhafte oder eine korrekte Nachricht. Darüber lernt der Angreifer, ob das von ihm manipulierte Padding korrekt ist und damit ein Teil des Klartextes richtig erraten wurde. Obwohl wir zunächst Padding-Orakel-Angriffe als Anwendungsfall betrachten, kann unser Verfahren auch entwickelt werden, um andere Arten von Seitenkanalangriffen zu erkennen, z. B. Angriffe, die Timing-Informationen ausnutzen.

Unser Verfahren ist in zwei Teile gegliedert, wie in Abbildung 2 dargestellt ist: Zuerst wird (1) ein Merkmalssatz aus Log-Ausgaben und Netzwerkverkehr (Netzwerkmitschnitte) erzeugt, der dann (2) vom maschinellen Lernverfahren genutzt wird, um zu bewerten, ob der TLS-Server durch einen Padding-Orakel-Angriff angreifbar ist oder nicht. Ist das Ergebnis, dass der TLS Server angreifbar ist, können wir mit Sicherheit sagen, dass er durch einen Padding-Orakel-Angriff angreifbar ist. Bekommen wir das Ergebnis, dass der TLS-Server nicht angreifbar ist, so gilt das nur für die Seitenkanäle, für die wir die Tests durchgeführt haben. Wir können also nur mit Sicherheit sagen, dass er nicht für Padding-Orakel-Angriffe angreifbar ist, aber keine Aussagen zu anderen Arten von Seitenkanälen machen.

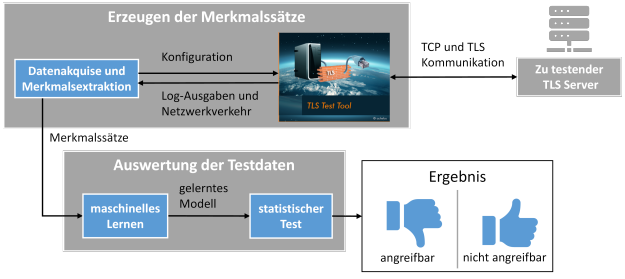

Abbildung 1: Schematische Darstellung unseres Verfahrens

Zum Erzeugen der Netzwerkmitschnitte nutzen wir das TLS Test Tool der achelos. Das TLS Test Tool ist ein TLS-Client, der so manipuliert wurde, dass er nicht nur normales, sondern auch fehlerhafter Verhalten implementiert oder Angriffe ausführen kann. Bei der achelos ist das TLS Test Tool Bestandteil des TLS Inspectors, eine Sammlung von Testwerkzeugen zum Testen und Zertifizieren von TLS-Implementierungen. Zum Erzeugen des Netzwerkmitschnitte wird das TLS Test Tool so gesteuert, dass ein Padding-Orakel-Angriff simuliert wird.

Ein Netzwerkmitschnitt besteht aus TCP- und TLS-Nachrichten, die zwischen einem TLS-Client und dem zu testenden TLS-Server ausgetauscht werden. Das Test Tool selbst ist in der Lage, manipulierte TLS-Nachrichten zu senden, wie auch ein Angreifer sie an den zu testenden TLS-Server senden würde. Um den Netzwerkmitschnitt zu erzeugen, versucht das Test Tool wiederholt TLS-Verbindungen zum TLS-Server herzustellen. Dabei werden durch unterschiedliche Konfigurationen des Test Tools unterschiedliche Manipulationen verwendet. Die Netzwerkkommunikation zwischen beiden Kommunikationspartnern wird aufgezeichnet und so der Netzwerkmitschnitt erzeugt.

Zur Verarbeitung der Netzwerkmittschnitte durch maschinelle Lernverfahren werden aus jedem Netzwerkmittschnitt 88 reellwertige Merkmale extrahiert. Diese sogenannten Merkmalssätze dienen als Eingabe für die maschinellen Lernverfahren.

Im Folgenden unterscheiden wir beobachtbare Informationen und geheime Informationen. Beobachtbare Informationen sind TCP-Nachrichten und verschlüsselte TLS-Nachrichten sowie andere Nachrichten auf anderen Ebenen des OSI-Schichtenmodells, die unser Verfahren nicht betrachtet. Geheime Informationen sind Teile der verschlüsselten TLS-Nachrichten im Klartext wie zum Beispiel das Master Secret, auf dessen Basis alle Schlüssel für die Verschlüsselung berechnet werden, sowie die eigentlichen zu übertragenden Daten wie zum Beispiel Passwörter, Login-Daten für Bankkonten oder wichtige Betriebsgeheimnisse.

Die Auswertung der Testdaten besteht wiederum aus zwei Komponenten: dem maschinellen Lernen und dem statistischen Test. Mit Hilfe des maschinellen Lernens versuchen wir binäre Klassifikatoren zu lernen, die Verbindungsversuche in zwei Klassen einteilen: (1) Verbindungsversuch mit korrektem Padding und (2) Verbindungsversuch mit manipuliertem Padding. Ein solcher Klassifikator entspricht also genau einem Padding-Orakel. Mit Hilfe eines statistischen Tests validieren wir das Ergebnis: Wenn ein gelernter Klassifikator in der Lage ist, die Korrektheit des Paddings mit einer Genauigkeit vorherzusagen, die statistisch signifikant besser als Zufall ist, haben wir ein Padding-Orakel gefunden und wissen damit, dass die TLS-Implementierung durch Padding-Orakel-Angriffe angreifbar ist.

3 Transfer in die praktische Anwendung

Unser Verfahren wird zum einen durch die Veröffentlichung einer kostenlosen, quelloffenen Variante sowie durch die Integration in den TLS Inspector der achelos in die Anwendung gebracht.

[5]Padding bezeichnet das Auffüllen eines Datenbestands, so dass bestehende Daten in eine Form gebracht werden, wie sie von einem Algorithmus benötigt werden.

Im Folgenden erklären wir die Integration in ein Produkt.

Unser Verfahren wurde in Form von einem Testfall prototypisch in den TLS Inspector integriert. Mit den aktuellen Ergebnissen sind wir bereits in der Lage, einen Testfall zu implementieren, der überprüft, ob ein TLS-Server durch Padding-Orakel-Angriffe angreifbar ist. Im weiteren Verlauf des Projekts werden weitere Testfälle hinzukommen, die TLS-Server auf die Angreifbarkeit von weiteren Arten von Seitenkanälen überprüfen.

3.1 Technische Umsetzung

Abbildung 3.1 zeigt ein Konzept für die technische Umsetzung des im Projekt entwickelten Verfahrens zur Integration in die den TLS Inspector. Über einen Testfall wird spezifiziert, auf welchen Seitenkanal geprüft und damit, welche Netzwerkmitschnitte benötigt werden. Der Testfall konfiguriert entsprechend der benötigten Netzwerkmitschnitte ein Modul für Datenakquise, das daraufhin die benötigten Netzwerkmitschnitte erzeugt. Für die Erzeugung der Daten kommt das TLS Test Tool zum Einsatz, welches prototypisch um die Manipulationen erweitert wurde, die für einen Padding-Orakel-Angriff benötigt werden. Das Test Tool wird vom Modul für Datenakquise so konfiguriert, dass es den gewünschten Angriff nachstellt. Die vom Test Tool gesendeten Nachrichten und die Antworten des zu testenden TLS-Servers werden aufgezeichnet und bilden den Datensatz als Eingabe für das Modul für maschinelles Lernen.

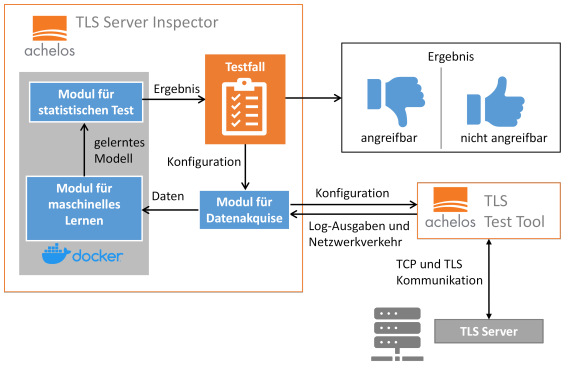

Abbildung 2: Technische Umsetzung

Der Testfall und das Modul für Datenakquise werden als Komponenten in die Testsuite integriert. Das Modul für maschinelles Lernen sowie das Modul für statischen Test werden in Form eines Docker Containers in die Testsuite integriert, da deren Konfiguration sehr aufwändig ist uns es sonst später in der Produktversion schwierig wäre, diese Kunden geeignet zur Verfügung zu stellen.

Literatur

[1] Jan Peter Drees, Pritha Gupta, Eyke Hüllermeier, Tibor Jager, Alexander Konze, Claudia Priesterjahn, Arunselvan Ramaswamy, and Juraj Somorovsky. Automated detection of side channels in cryptographic protocols: Drown the robots! In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, AISec '21, pages 169–180, New York, NY, USA, 2021. ACM.