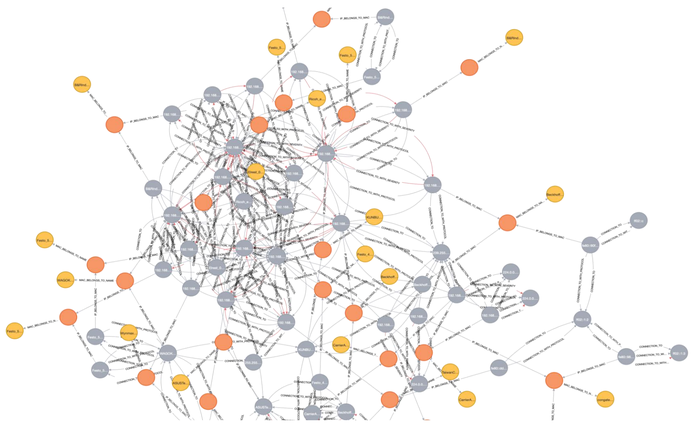
Digitale Testlandschaft für die APT Abwehr auf Basis von Digitalen Zwillingen

Praxisnah: Brownfield-First Reallabor der SmartFactoryOWL

OPC UA · MQTT · PROFINET · HTTP · EventLog · /var/log/* · Modbus TCP · SSH · DHCP · ICMP · RDP · Telnet · DNS · SMB · NTP · ARP · IGMP · SSDP

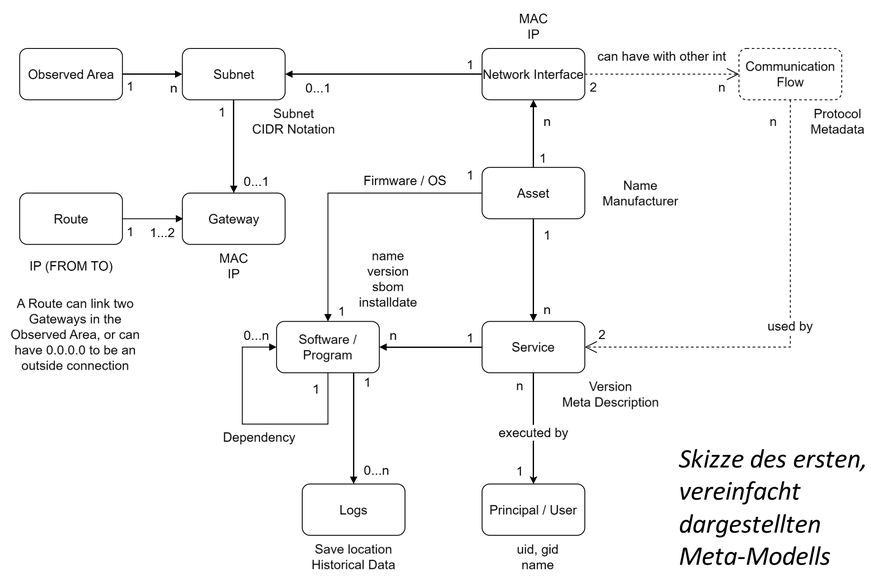


DZ-Pipeline

Informationen (passiv) sammeln

No.	Time	Source	Destination	Protocol	Length	Info
25	0.082649	192.168.0.15	192.168.0.10	OpcUa	139	UA Secure Conversation Message: PublishResponse
27	0.095184	192.168.0.34	192.168.0.31	OpcUa	143	UA Secure Conversation Message: PublishResponse
29	0.095876	192.168.0.31	192.168.0.34	OpcUa	120	UA Secure Conversation Message: PublishRequest
31	0.096772	192.168.0.34	192.168.0.31	TCP	64	4840 → 59229 [ACK] Seq=86 Ack=67 Win=59934 Len=0
33	0.097725	192.168.0.34	192.168.0.31	TCP	64	[TCP Window Update] 4840 → 59229 [ACK] Seq=86 Ack=67 Win=59942 Len=0
35	0.097853	192.168.0.34	192.168.0.31	TCP	64	[TCP Window Update] 4840 → 59229 [ACK] Seq=86 Ack=67 Win=60000 Len=0
37	0.111758	192.168.0.10	192.168.0.15	OpcUa	124	UA Secure Conversation Message: PublishRequest
39	0.111945	192.168.0.15	192.168.0.10	TCP	60	4840 → 57552 [ACK] Seq=86 Ack=67 Win=59934 Len=0
41	0.112377	192.168.0.15	192.168.0.10	TCP	60	[TCP Window Update] 4840 → 57552 [ACK] Seq=86 Ack=67 Win=59942 Len=0
43	0.112511	192.168.0.15	192.168.0.10	TCP	60	[TCP Window Update] 4840 → 57552 [ACK] Seq=86 Ack=67 Win=60000 Len=0
45	0.146111	MicrochipTec_2b:ab:d5	00:00:00_00:00:00	ARP	64	Who has 192.168.0.1? Tell 192.168.0.37
46	0.146111	MicrochipTec_2b:ab:d5	00:00:00_00:00:00	ARP	60	Who has 192.168.0.1? Tell 192.168.0.37
47	0.146111	MicrochipTec_2b:ab:d5	00:00:00_00:00:00	ARP	64	Who has 192.168.0.1? Tell 192.168.0.37
48	0.157752	192.168.0.233	192.168.0.4	TCP	126	52236 → 11740 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=68

Datenaggregation

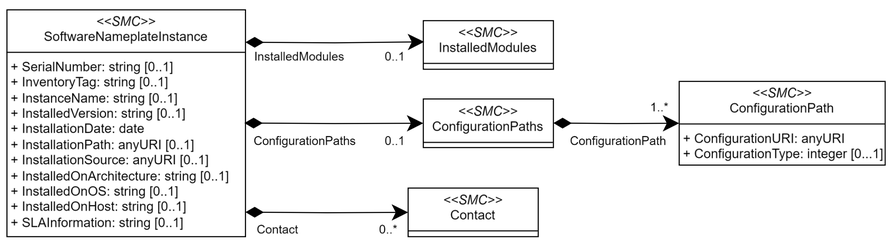


Skizze des ersten, vereinfacht dargestellten Meta-Modells

Digitaler Zwilling

Asset Application Shell (AAS), zu deutsch: Verwaltungsschale Standardisiertes Konzept zur Modellierung von Digitalen Zwillingen

[2]



Bsp. für die Verwaltungsschale – IDTA Software Nameplate [1]

Container Orchestrator

Mittels Ollama werden von uns Large Language Models (LLMs) verwendet, um aus Verwaltungsschalen sogenannte Docker-Compose Dateien zu erstellen. Dies sind „Baupläne" für Container-Ökosysteme. Dieser Methode wird ein selbstgeschriebener Parser für Verwaltungsschalen gegenübergestellt und evaluiert.

[4]

Container Ökosystem der digitalen Testlandschaft

[3]

Testlandschaft

Vulnerability Scans etc., Austesten, Training, Anomalieerkennung

Problemstellung

In industriell genutzten Netzwerken agieren Systeme mit diversen Betriebssystemen, Protokollen und Sicherheitsrichtlinien. Angriffe können Auswirkungen auf unterschiedliche Komponenten haben. Es gibt somit keine alleinige Datenquelle, aus der alle Angriffe erkannt werden können. Wir benötigen daher die Kombination diverser Datenquellen (Netzwerkverkehr, Systemprotokolle, Sensordaten), um diverse Angriffsmuster zielführender erkennen zu können. Das Training von Algorithmen zur Erkennung dieser Angriffsmuster profitiert von realen Datenquellen.

Relevanz der Lösung

Angriffssimulationen in realen Netzwerken einer Produktionsumgebung sind zwar theoretisch möglich, bringen aber erhebliche Nachteile mit sich. Kritische Verfahren zur Absicherung von Produktionsanlagen können dadurch nicht ohne Risiko getestet werden, da die eingesetzten Komponenten potenziell gefährdet werden könnten. Zudem stellen bestehende Sicherheitssysteme wie Endpoint Detection and Response-Lösungen weitere Herausforderungen dar. Digitale Zwillinge bieten hier eine innovative Alternative, indem sie sichere und flexible Testumgebungen für derartige Szenarien schaffen.

Datenaggregation

Netzwerkverkehr und Logs eines definierten Netzwerks werden durch passive TAPs und Log-Aggregation zentral gesammelt und zugänglich gemacht. Damit die aufgezeichneten Daten für die APT-Erkennung nutzbar werden, müssen sie den Aktivitäten des Systems zugeordnet werden. Dazu muss ein semantischer Zusammenhang zwischen den Datenpunkten ermittelt und ein Modell des betrachteten Netzwerkbereiches aufgebaut werden. Das initiale Modell wird durch die Erfassung von Daten kontinuierlich erweitert und fortgeschrieben. Zudem werden im Meta-Modell Referenzen zu den genutzten Datenquellen gespeichert. Dieses Modell dient nachfolgend als Basis für die Erstellung der digitalen Zwillinge.

Digitaler Zwilling

Aus der Datenaufbereitung werden segmentierte und semantisch zusammenhängende Daten bereitgestellt, welche in digitale Zwillinge umgewandelt werden. Digitale Zwillinge sind nach dem Verständnis der Industrie bislang Modelle, welche Objekte und deren Zusammenhänge aus der realen Welt wiederspiegeln. Für diverse Gegebenheiten sind bereits standardisierte Teilmodelle durch Organisationen wie der IDTA [1] erstellt worden. So kann beispielsweise ein Interface einer Maschine oder Software standardisiert beschrieben werden. Im Rahmen des Projektes werden Vorschläge für die fehlenden Teilmodelle sowie die Erweiterung der bestehenden erarbeitet, um sowohl die bestehenden Lücken zu schließen als auch eine universelle Schnittstelle für das Threat Hunting oder andere Security-Anwendungen zu schaffen.

Digitale Testlandschaft

Unterstützt durch generative KI und Netzwerkemulation wird aus der Modellbeschreibung des Netzwerkes die Konfiguration eines lauffähigen Abbildes aller Komponenten für einen Container Orchestrator erzeugt. Indem wir eine bestehende Systemlandschaft möglichst genau replizieren, können wir eine Testumgebung für Angriffssimulationen bereitstellen, welche unabhängig von der realen Umgebung agiert. Wir stellen dabei Netzwerke, Recheneinheiten (Computer, Steuerungen, …) sowie von uns erkannte Services dar, welche auf den Systemen laufen. Es ist geplant, aus dem Netzwerk-Traffic Beispieldaten zu extrahieren, welche von den Services ohne deren interne Logik in der Testlandschaft kommuniziert werden können.

Projektkonsortium

inIT | TH OWL · Data|H · rt-solutions.de experts you can trust.

Kontaktdaten

Robin Foster, M.Sc. | Dipl.-Math. Natalia Moriz | Prof. Dr.-Ing. Henning Trsek

inIT – Institut für industrielle Informationstechnik

Technische Hochschule Ostwestfalen-Lippe

Campusallee 6 | 32657 Lemgo

{robin.foster; natalia.moriz; henning.trsek}@th-owl.de

Referenzen

[1]: https://github.com/admin-shell-io/submodel-templates

[2]: https://github.com/eclipse-aaspe/package-explorer

[3]: https://www.docker.com

[4]: https://github.com/ollama/ollama

Diverse Icons: Adobe Stock – lizensiert für Bildungseinrichtungen

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung