EP SOTERN IMT-Atlantique – IRISA

Suivi automatique d'attaquant dans un SI et adaptation temps-réel du système pour lui faire dévoiler ses TTP

Gilles Guette
EP SOTERN, Rennes, France
gilles.guette@univ-rennes1.fr

Mots Clés : Cybersécurité, Deceptive Platform, Threat Hunting

Contexte

Avec la multiplicité des systèmes connectés à Internet et la dépendance vitale des entreprises à leur système d'information, les attaquants et les défenseurs se sont adaptés au progrès techniques, méthodologiques et organisationnels de chacun. Du côté offensif, on observe un accroissement de nouvelles menaces de plus en plus sophistiquées et subtiles. Les APT (Advanced Persistent Threat) [1] sont des attaques se poursuivant sur une longue période de temps afin d'en augmenter l'impact et la furtivité et qui s'adaptent aux capacités du défenseurs. Ces attaques sont souvent menées par des groupes d'attaquants professionnels très compétents et dotés de moyens conséquents, parfois subventionnés par un état.

Du côté défensif, les outils et techniques de détection et de *Threat Hunting* ont aussi évoluées [2] avec le développement des SOAR (Security Orchestration, Automation and Response) [3], des EDR/XDR, SOC, SIEM [4]...Tous ces moyens ont été développés pour de la réponse à incidents sur des systèmes en production. C'est-à-dire pour détecter au plus tôt une attaque et réagir au mieux. Un certain nombre de ces outils bien qu'efficace reste limité à de la détection sur règles et à de la réaction par isolement de l'attaquant.

Nous nous intéressons à la compréhension de l'attaque et des attaquants. C'est pourquoi nous travaillons sur les *deceptive platform* aussi autrefois appelées *honeynet* [5, 6]. Un *honeynet* est un réseau factice développé pour dérouter un attaquant (le détourner de son objectif primaire) ou pour étudier ses TTP (Tactiques, Techniques et Procédures d'attaque [7]). Un honeynet est la représentation d'un système d'information classique d'entreprise volontairement conçu avec des failles, des vulnérabilités exploitables ou des faiblesses de conception. Le but de ces systèmes est, s'ils sont directement connectés à l'Internet, d'attirer les attaquants pour qu'ils puissent jouer leurs attaques. En ce sens un *honeynet* peut être vu comme un cyber range [8] pour attaquant. S'ils sont connectés au réseau d'entreprise, ils peuvent aussi être utilisés pour détourner l'attaquant du SI réel afin de l'occupé et de l'isoler pour avoir le temps de réagir. Ce réseau factice vulnérable a donc intérêt à être le plus réaliste possible afin d'être plus attractif que le SI de l'entreprise.

Nous nous intéressons à la génération paramétrée de *deceptive platform*. Des travaux précédents et en cours s'intéressent à la modélisation de l'attaquant [9] et à la génération automatique d'infrastructures vulnérables [10] à partir de scénarios d'attaque pertinents et des contraintes d'une infrastructure physique. L'idée est à partir de la description d'un scenario d'attaque impliquant plusieurs machines d'un système d'information de générer automatiquement ce SI dans lequel l'attaque est jouable.

Objectifs

Les objectifs de cette thèse sont d'explorer les aspects surveillance et réactivité/adaptabilité d'une telle infrastructure.

Surveillance

Le premier objectif est de savoir ce qu'il faut observer, où et comment de la manière la plus discrète possible pour pouvoir suivre l'attaquant dans sa progression dans le système. Une fois que l'environnement est modélisé avec tous ses éléments constituants, avant de le mettre en place, il faut être en mesure de pouvoir surveiller toutes les activités que l'ont pourrait attribuer à l'attaquant. Lorsque l'architecture générée, l'est dans le but d'attirer un attaquant, les vulnérabilités constituant le scénario d'attaque envisagé sont connues. Un certain nombre de sondes et de dispositifs de surveillance peuvent donc être mis en place pour collecter toutes les données pertinentes. Néanmoins à ce stade beaucoup de difficultés restent présentes et beaucoup de questions scientifiques se posent :

- Comment *s'assurer* qu'il n'existe pas d'autres vulnérabilités exploitables dans l'architecture générée ? En effet, aucun système n'est sûr et il est raisonnable de considérer que des attaques dites *zero-day* (attaque connue seulement par l'attaquant qui vient de la découvrir et par conséquent ne disposant pas de correctif ni de règle de détection) existent sur les systèmes utilisés.
- Comment garantir et prouver la couverture de surveillance ? Les systèmes sont devenus tellement complexe, que prouver que la solution de surveillance mise en place ne rate aucun chemin d'attaque possible ni aucun évènement n'est pas trivial. De plus, l'existence de *zero-day* étant actée, comment prouver que même en cas d'évènements manqués par la surveillance, l'ensemble des choses capturées est suffisante pour détecter l'attaque et pour comprendre ce qu'on a raté ?
- Comment surveiller de manière *furtive* ou *naturelle* afin de ne pas effrayer l'attaquant ? Évidemment, l'attaquant ayant pris pieds dans notre système, il va être en mesure de découvrir les différentes activités sur celui-ci. Donc soit il ne doit pas être en mesure de détecter le système de surveillance (et il faut prouver cette propriété) soit la connaissance du système de surveillance ne doit pas l'effrayer.
- Comment surveiller de manière efficace sans générer une quantité de log inutilisable ? En effet, la problématique classique des solutions de supervision système et réseau est la quantité de log générés pouvant aller jusqu'à plusieurs TeraOctets/heure.
- Surveiller le « faux SI » est-il suffisant ? En effet, l'infrastructure générée est une infrastructure virtuelle, elle repose donc sur un environnement matériel et un hyperviseur qui va gérer ces différentes machines virtuelles. Il faut donc aussi garantir que l'attaquant ne peut pas profiter d'une faille pour sortir de la VM et sauter dans l'hyperviseur pour en prendre le contrôle et ainsi devenir potentiellement maître de toutes l'architecture proposée.

Les réponses à ces questions passent par une connaissance fines des systèmes ainsi qu'une modélisation du fonctionnement des attaques. **La méthodologie de validation de telles solutions est aussi une question ouverte.** Demander à de véritables attaquants (malveillants) d'attaquer le système en utilisant toutes leurs techniques et tous leurs outils n'est pas une option envisageable.

Adaptation/réaction

Suite à ce premier objectif de surveillance, le second but et de pouvoir réagir à la volée en fonction de ce qu'on observe du comportement de l'attaquant en modifiant ou en complétant l'infrastructure vulnérable afin de forcer l'attaquant à dévoiler un maximum de ses capacités.

Par exemple, lui couper l'accès à une ressource pour voir s'il sait revenir différemment ou désactiver un compte qu'il utilise pour voir si il en contrôle d'autres.

Le but est de pouvoir réagir à la volée en fonction de ce qu'on observe du comportement de l'attaquant en modifiant ou en complétant l'infrastructure vulnérable. L'objectif étant de caractériser l'attaque et l'attaquant et de le forcer à dévoiler un maximum de ses capacités afin d'augmenter notre connaissance de l'arsenal offensif existant.

La réaction aux attaques peut être effectuée de plusieurs manières. La première et la plus évidente est de façon manuelle par un opérateur interagissant en temps réel avec le système. Il est assez évident que ce n'est pas tenable dans le temps. Nous nous orientons donc dans un premier temps vers deux pistes. La première est une manière automatique et statique. C'est-à-dire qu'en fonction du scénario déployé ou du type d'attaque anticipée, un certain nombre de contre mesures ont été implémentées dans un *agent de contrôle* et celui-ci déclenche les réponses en fonction des événements observés. Cette première approche peut servir de preuve de concept montrant qu'il est possible de faire évoluer dynamiquement le système tout en gardant ses propriétés et en modifiant au besoin les règles de surveillance.

La seconde approche envisagée est beaucoup plus autonome et par conséquent devrait être prouvée avant une utilisation complète. Il s'agit de l'utilisation d'algorithmes d'IA pour apprendre le comportement du système sain (ou le comportement d'un attaquant) et les réponses appropriées. Cela lève donc les challenges du volume de données nécessaire à l'apprentissage en cas d'utilisation de machine learning. Si ces données ne sont pas disponibles, est-il possible d'utiliser des algorithmes d'IA qui s'entraînent seuls comme du reinforcement learning (et dans notre cas surtout du Adversarial Machine/Reinforcement Learning avec un humain en face) et comment prouve-t-on que l'agent attaquant qui interagira avec l'outil de défense (resp. que l'agent de défense avec qui l'attaquant interagira) est correct. Et autre point et certainement pas le dernier, comment garantit-on, que les outils d'IA utilisés ne font pas pire que l'attaquant ?

Approche envisagée

Le ou la candidat.e recruté.e pourra se servir des travaux initiés dans une thèse de l'équipe SOTERN permettant de générer un scénario d'attaque puis l'infrastructure virtuelle nécessaire au rejeu du scenario afin de pouvoir étudier sur des cas réels la pertinence de la surveillance et son efficacité afin de créer une modélisation réaliste.

Par rapport à un système réel sous attaque, dans un *honeynet* le premier point (la surveillance) pourrait de prime abord sembler plus facile car on connaît le scenario d'attaque. Cela est partiellement vrai, effectivement ce qui est connu vulnérable doit être surveiller, mais rien n'empêche l'attaquant d'essayer d'autres vecteurs ou d'utiliser des procédures non connue pour le moment. L'un des points essentiels de la thèse est donc d'explorer la possibilité de formalisation/modélisation du processus de surveillance du SI avant d'envisager la réaction.

Environnement de recherche

La thèse se déroulera à Rennes sur le campus de d'IMT-Atlantique au sein de l'équipe SOTERN, à partir de septembre/octobre 2024 pour une durée de 3 ans. Il/elle sera encadré.e par Gilles Guette impliqué dans ce projet. Cette thèse se déroule dans le cadre du projet PEPR Cybersécurité SuperViz (https ://files.inria.fr/superviz/).

Compétences attendues

Les compétences techniques requises pour cette thèse sont de bonnes connaissances en **systèmes (windows/Linux), réseaux et sécurité** un bon relationnelle, une capacité à s'exprimer à l'oral et à l'écrit en langue anglaise.

Candidatures

Le démarrage de la thèse étant lié à un processus administratif strict, les candidatures accompagnées d'un CV et d'une lettre de motivation sont à envoyer le plus tôt possible par mail à **gilles.guette@univ-rennes.fr**.

Références

[1] R. S. Ross. Managing Information Security Risk : Organization, Mission, and Information System View - NIST. In *Special Publication (NIST SP)*, 2011.

[2] Boubakr Nour, Makan Pourzandi, and Mourad Debbabi. A survey on threat hunting in enterprise networks. *IEEE Communications Surveys & Tutorials*, 2023.

[3] Le SOAR qu'est-ce que c'est ? https ://www.redhat.com/fr/topics/security/what-is-soar, 2022.

[4] SOC SIEM XDR MDR EDR... quelles différences ? https ://www.orangecyberdefense.com/fr/insights/blog/detection/soc-siem-xdr-mdr-edr-quelles-differences, 2023.

[5] Vincent Nicomette, Mohamed Kaâniche, Eric Alata, and Matthieu Herrb. Set-up and deployment of a high-interaction honeypot : experiment and lessons learned. *Journal in Computer Virology*, 7(2) :143–157, May 2011.

[6] The Honeynet Project. https ://www.honeynet.org, 2017.

[7] The MITRE Corporation. The MITRE ATT&CK Matrix for Enterprise. https://attack.mitre.org/matrices/enterprise, 2018.

[8] Muhammad Yamin, Basel Katt, and Vasileios Gkioulos. Cyber ranges and security testbeds : Scenarios, functions, tools and architecture. *Computers & Security*, 88 :101636, 10 2019.

[9] A. Berady, V. Viet Triem Tong, G. Guette, C. Bidan, and G. Carat. Modeling the Operational Phases of APT Campaigns. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 96–101, 2019.

[10] Pierre-Victor Besson, Valérie Viet Triem Tong, Gilles Guette, Guillaume Piolle, and Erwan Abgrall. Ursid : Using formalism to refine attack scenarios for vulnerable infrastructure deployment, 2023.