Eindopdracht Python: Trojan (of Agent) Framework

Ontwikkel een flexibele Trojan-agent met GitHub-integratie

Doel: je ontwikkelt een modulaire Trojan-toepassing die communiceert met een GitHub-repository om opdrachten op te halen en uit te voeren. Het project combineert technische kennis met een ethische reflectie, waarbij je inzicht krijgt in de werkwijzen van hackers en hoe deze technieken passen binnen hun modus operandi. Naast de verplichte basisfunctionaliteiten ontwerp je (minstens) drie custom modules die specifieke (ethische) hacking-acties uitvoeren.

Belangrijk!

- Het project is puur educatief bedoeld. Test uitsluitend in een virtuele omgeving of sandbox.
- Deel geen inloggegevens of gevoelige informatie.
- Wees je bewust van de ethische implicaties van het project.
- Denk zelf na en gebruik de vrijheid om er iets innovatief van te maken.
- Zorg dat de inspanning jouw inspanning is. Plagiaat = a deadly sin. ChatGPT gebruiken op een ethische manier.
- Reflecties steeds in Markdown.

Week 1: Analyse en Conceptvorming (indienen uiterlijk 4 december)

Doel: begrijp de onderliggende fenomenen en de rol van een Trojan binnen de modus operandi van hackers.

Deelopdrachten:

1. Theoretische analyse:
    - Onderzoek de belangrijkste doelen van hackers bij het inzetten van een Trojan:
    - Waarom en hoe worden Trojans gebruikt?
    - Wat is het verschil tussen een backdoor, een RAT (Remote Access Trojan), en een agent?
    - Hoe past dit in een breder kader van hacking, zoals reconnaissance, privilege escalation, en exfiltratie?
    - Documenteer je bevindingen in een verslag (max. 1 A4) en bespreek ethische overwegingen bij het gebruik van dergelijke technieken.
2. Technische analyse:
    - Maak een overzicht van hoe je Trojan (of Agent) er qua opbouw zal uitzien en van mogelijke modules die een Trojan kan uitvoeren.
    - Kies minimaal drie custom modules en licht toe:
    - Welke functionaliteit hebben deze modules?
    - Waarom kiezen hackers deze technieken?
    - Hoe kunnen ze worden gebruikt in een ethical hacking context?

Week 2-4: Ontwikkeling van de Trojan (tussenstanden indienen op 11 en 18 december)

Basisfunctionaliteit (verplicht):

1. GitHub-Repository:
    - Structuur:
        - config: Voor configuratie-informatie (zoals te runnen modules en instellingen).
        - data: Voor de verzamelde resultaten van modules.
        - modules: Voor de modulecode (uitbreidbaar met nieuwe acties).
    - Configuratiebestand (config.json):
        - Bestuurt je Trojan met een lijst van modules die uitgevoerd moeten worden (is leeg indien slapend).
        - Hou rekening met een botnet en gebruik unieke ID's per client.
    - Data-exfiltratie:
        - Verzamelde data wordt teruggestuurd naar de data-directory in het repository.
2. Trojan-framework:
    - Verbind met het GitHub-repository via de GitHub API (of geschikte Python library naar keuze).
    - Download en voer modules uit volgens de configuratie.
    - Gebruik een custom importer om modules te laden vanuit het GitHub-repository.
    - Randomize de frequentie van pollen van de configuratie (denk hier zelf na wat het slimst is) om detectie te voorkomen.

Custom Modules (3 verplicht):

Ontwerp minimaal drie modules die passen binnen een ethical hacking context. Hier zijn enkele suggesties:

1. DDoS-module:
    - Voer een gesimuleerde DDoS-aanval uit op een doelwit via HTTP-requests of UDP-pakketten.
    - Reflecteer op hoe dit in een realistisch scenario zou worden gebruikt en welke impact het heeft.
2. Portscan-module:
    - Scan de poorten van een opgegeven IP-adres en rapporteer open poorten.
    - Combineer dit met reconnaissance-methodologie.
3. Keylogger-module:
    - Registreer toetsaanslagen op een veilige, gecontroleerde testmachine.
    - Reflecteer op de ethische implicaties van het verzamelen van dergelijke gegevens.
4. Backdoor-module:
    - Maak een TCP- of HTTP-backdoor waarmee je via een remote verbinding eenvoudige commando's op de besmette host kunt uitvoeren.
    - Voeg een inactiviteitstimer toe om misbruik te beperken.
5. System Sniffer:
    - Monitor netwerkverkeer op de besmette host en log HTTP-verzoeken naar het GitHub-repository.
    - Gebruik dit om inzicht te krijgen in het verkeer.

6. System Enumeration Module:
    - Verzamel informatie over gebruikersaccounts, draaiende processen, en systeeminformatie.
    - Gebruik dit als een eerste stap in privilege escalation.

Alstublieft: verras je docent met innovatieve eigen keuzes 😉

Week 4: Afwerken, testen en eindreflectie (indienen uiterlijk 20 december)

Doel: Werk je opdracht af, test je Trojan (of Agent) en reflecteer over het ethische gebruik ervan.

Opdrachten:

1. Werk je product af en test alle modules in een veilige en gecontroleerde omgeving. Documenteer je bevindingen.
2. Reflecteer over:
    - Welke technieken het meest effectief waren.
    - Hoe dit in een realistisch scenario misbruikt kan worden en hoe je zulke aanvallen kunt voorkomen.
    - Jouw eigen leerproces tijdens de opdracht.

Eindproducten

1. Code:
    - Een volledig werkende Trojan met:
        - Basisfunctionaliteit.
        - Minstens drie custom modules.
        - Gebruik een virtual environment (venv) en een requirements.txt bestand voor afhankelijkheden. Venv-folder niet mee opladen!
2. GitHub-repository:
    - Structuur volgens de basisfunctionaliteit.
    - Gehost op een private repository, inclusief configuratie, modules en data (geef toegang aan Github-gebruiker admkrm, kristof.michiels@gmail.com)
3. Reflectieverslag:
    - Beschrijf je bevindingen, keuzes, uitdagingen, en ethische inzichten.
    - Demo: demonstreer de werking van je Trojan met een video van max 3min en licht je reflectieverslag toe.

Alternatief voor Systeembeheer: Agent Framework

Voor studenten die liever een systeembeheergerichte benadering kiezen in plaats van een Trojan te ontwikkelen, is het mogelijk om een Agent Framework te bouwen. Dit framework wordt ontworpen om vrijwillig geïnstalleerd te worden op systemen, zoals servers of werkstations, met als doel geautomatiseerde systeembeheer- en monitoringstaken uit te voeren.

Specifieke Doelstellingen:

1. Systeembeheerfunctionaliteit:

- Voer periodieke systeemcontroles uit, zoals het controleren van schijfruimte, het monitoren van CPU- en RAM-gebruik, en het genereren van systeemlogs.
- Verzamel en rapporteer gegevens naar een centrale GitHub-repository.
- Mogelijkheid om op afstand opdrachten uit te voeren, zoals software-updates of logbestanden verzamelen.

2. Veiligheid en Controle:

    - Alle acties zijn transparant en vrijwillig.
    - Gebruikers kunnen zelf configureren welke acties de agent uitvoert via een config.json.

3. Custom Modules (3 verplicht):

- Bijvoorbeeld:
    - Health Monitoring Module: Bewaak CPU-gebruik, RAM, en netwerkactiviteit.
    - Backup Module: Automatiseer het maken van back-ups van een specifieke directory naar een cloudopslag.
    - Remote Command Execution Module: Laat beheerders commando's verzenden en uitvoeren via de GitHub-configuratie.

Uiteraard kan je hier vanuit het defensief perspectief ook security gerelateerde modules verzinnen! Bespreek met je docent.

Technische Overeenkomsten:

- Net als bij de Trojan integreert de Agent met een GitHub-repository om configuraties op te halen en data te rapporteren.
- Gebruik objectgeoriënteerde principes en zorg voor een uitbreidbaar framework.
- Focus ligt op monitoring en systeembeheer in plaats van hacking-methodologie.

Ethische Reflectie:

- Beschrijf hoe het framework een positieve impact kan hebben op IT-beheer.
- Reflecteer op de ethische verantwoordelijkheid bij het inzetten van agents op systemen van derden.