esse

Security for Systems Engineering – VO 08: Capture the Flag

Martin Moutran
Florian Fankhauser

INSO – Industrial Software
Institut für Information Systems Engineering | Fakultät für Informatik | Technische Universität Wien

Agenda

Grundlagen zum CTF-Contest

Organisatorisches

Aufbau der Übungsumgebung

Bewertung

Auszug Regeln

Tipps zur Vorgehensweise

Anhang

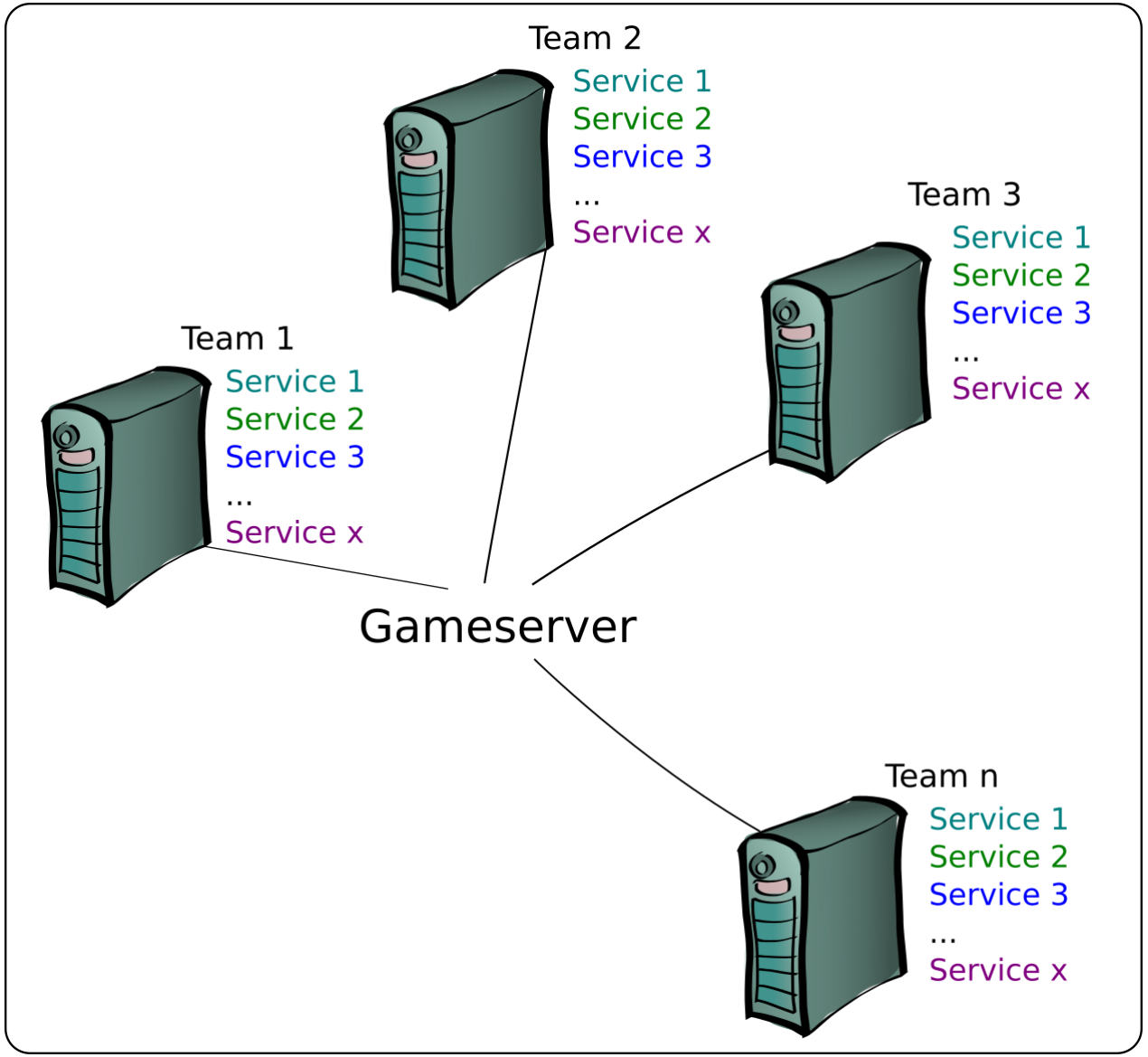
CTF – Capture The Flag

- CTF → Capture the Flag
  - Traditionell: Offline-Spiel
  - Teams versuchen jeweils die gegnerische Fahne zu entwenden und in das eigene Lager zu bringen
- In der LVA
  - Online-Spiel
  - Bewerb Teil der Übung

CTF in Security for Systems Engineering

- Teams treten gegeneinander an
- Spielfeld ist eine abgeschlossene Umgebung
- Flags sind Strings in Services
- Flags werden vom Gameserver verteilt
- Ziele für die Lehrveranstaltung
  - Praktische Erfahrungen in IT-Sicherheit
  - Erkennung und Behebung von Schwachstellen in Systemen
  - Ausarbeitung von Angriffstechniken (Sicherheitstests)
  - ... und natürlich viel Spaß ;-)

ESSE-CTF Contest Überblick



Voraussetzungen zur Teilnahme

- StudentIn von Security for Systems Engineering / IT Security in Large IT Infrastructures
- Teilnahme nur mit eigenem Notebook – ACHTUNG bei Team-Einteilung
- Anmeldung für CTF-Contest in TUWEL
- Kenntnisse
  - Inhalte Introduction to Security VU
  - Inhalte Security for Systems Engineering VU
  - Linux (siehe Linux Workshop Slides aus Introduction to Security)
  - Unterschiedliche Programmiersprachen

Vorbereitung Programmiersprachen

- Auffrischung von Programmierkenntnissen
- Behebung von Schwachstellen
  - Inputvalidierung
  - Korrektur von logischen Fehlern
- Beispiele für Programmiersprachen
  - C, C++, Java, PHP, Python, Ruby, Shell Scripts, ...
  - Vielleicht auch Exoten
    - `http://en.wikipedia.org/wiki/Esoteric_programming_language`
    - `http://esolangs.org/wiki/Language_list`

Paralleler Besuch zweier ESSE-LVAs

- Bei parallelem Besuch von IT Security in Large IT Infrastructures CTF-Contest-Ersatz verpflichtend
- Punkte des CTF können nur für IT Security in Large IT Infrastructures verwendet werden
- Bitte E-Mail an lva.security@inso.tuwien.ac.at

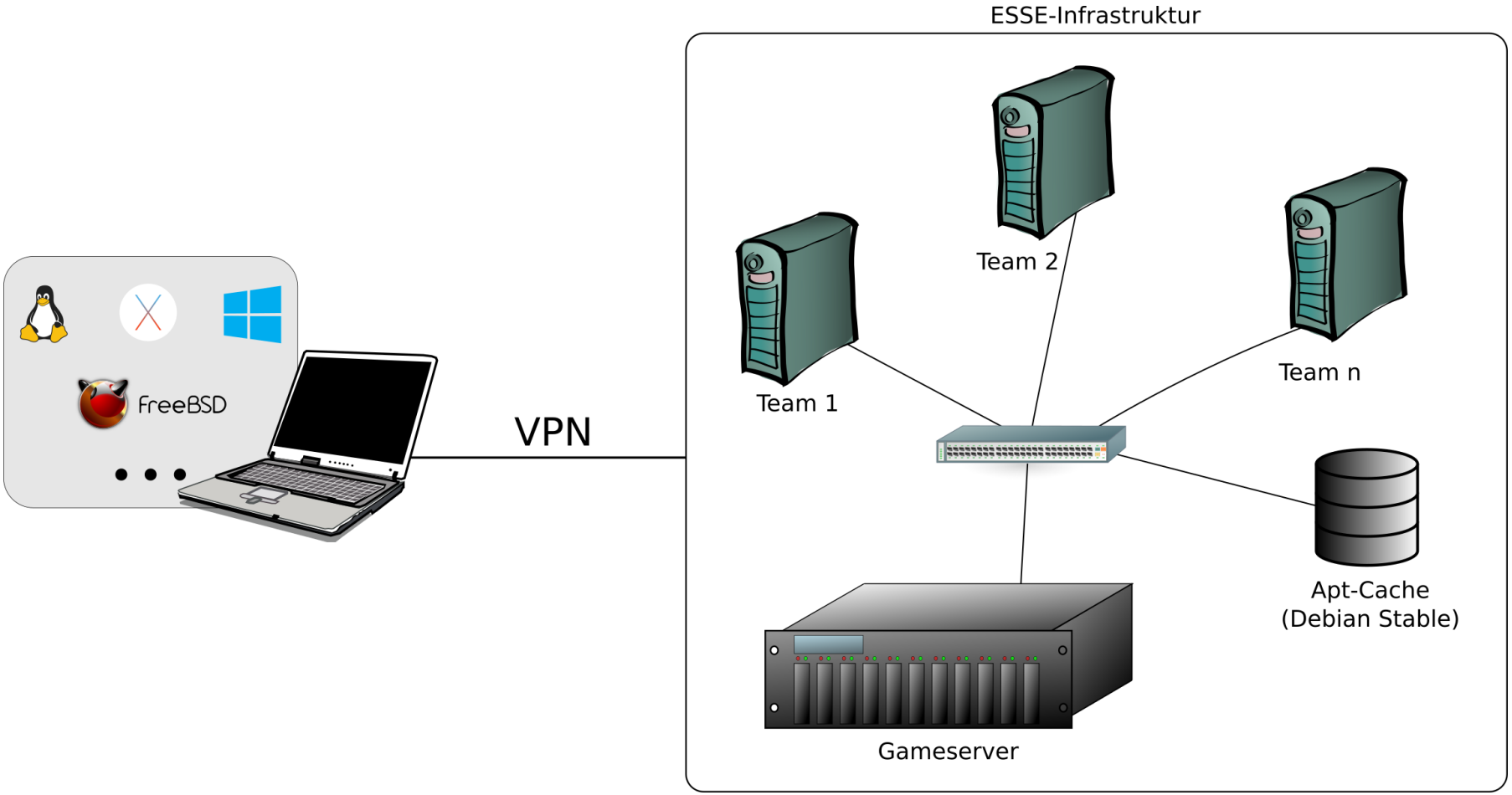
Termin

- 1 Termin mit fixen Plätzen
  - Sa. **09.06.2018**
  - Einlass ab 09:00 Uhr
  - Analyse-/Setup-Phase des Team-Servers: 09:30-10:30 Uhr
  - Bewerb: 10:30-16:30 Uhr
  - Preisverleihung ca. 17:00 Uhr
  - Protokoll bis 18:00 Uhr
- Treffpunkt für Einlass
  - Hörsaal 8 – TU Hauptgebäude → **nur unterer Eingang (EG)**
  - *Oberer Eingang → ESSE HQ*
- Anmeldung über TUWEL

Teambildung Security for Systems Engineering

- Teams zu je 4 Studierenden
- Fixe Anmeldung von Team-Namen: 01.06.2018 bis 07.06.2018 (TUWEL)
- Beispiele für kreative Namen bisheriger CTF-Contests
  - „sudo make me a sandwich“
  - „Schaf-256“
  - „it_hurts_when_IP“
- Beispiele für *mittlerweile* unkreative Namen
  - „'); DROP TABLE students; –“
  - „; DROP table groups –“
- Kein Team-Name registriert → wir vergeben *gerne* einen ;)
- Im Einzelfall behalten wir uns vor Team-Namen abzulehnen :)

Aufbau der Übungsumgebung – Übersicht



Aufbau der Übungsumgebung – Bestandteile

- Pro Team ein Server
- IP-Adressen der Server werden bekannt gegeben
- Gameserver
- Zentraler Knoten (VPN, Logging!)
- VPN
  - OpenVPN
  - Config + Anleitung wenige Tage vor Termin in TUWEL
  - Login: Lab0-Credentials
- Clients: Notebooks
  - ggf. Live-System auf USB-Sticks o.ä.
  - Vorab-Test der WLAN-Verbindung (eduroam, tunet) empfohlen

Team-Server

- Alle Teams bekommen (fast) identen Server
  - Unterschied bei IP-Adresse
- Login: Lab0-Credentials, `sudo` verfügbar
- Dienste sind selbstentwickelte Applikationen mit Schwachstellen
  - Schwachstellen sollen gefunden und behoben werden
  - Angriffe auf die Schwachstellen der anderen Team-Server
  - Beispiele für Arten von Sicherheitsfehlern
    - Implementierung (*-Injection, Logikfehler, ...)
    - Konfiguration (Standard-Passwörter, Berechtigungen, ...)
    - *Hinweis:* C bzw. C++ heißt nicht automatisch Buffer-Overflow!

Funktionsweise und Format von Flags

- Flags sind Daten der Dienste
- Periodische Verteilung durch Gameserver
- Flags besitzen eingeschränkten Gültigkeitszeitraum (ca. 15 Minuten)
- Format
  - [Timecode inkl. Zeitzone][Zufalls-String]
  - Beispiel: **0206201118045 0UTC**3ZL8T6XW1QKSJUU
  - Timecode entspricht Verfallsdatum
  - Achtung Test-Flags: 0206201118081 4UTC**TEST**970VUKCGZIF

Zeitverlauf

- Aktivierung der Zugänge (VPN, Team-Server)
- Analyse, Absicherung der Server ohne Bewertung
- Parallele Ausarbeitung des Protokolls für LVA-Bewertung
- Gameserver startet zeitverzögert mit der Bewertung
- CTF-Contest läuft
- Nach Spielende: SiegerInnenehrung und Protokollabgabe
- Pause(n): keine Vorgabe, Selbstorganisation

Bewertung

- Unterscheidung zwischen Bewerb und Punkte für Lab
  - Übung → lab2, Bewertung auf Basis des Protokolls
  - Bewerb → Spaß und Preise
- Security for Systems Engineering oder IT Security in Large IT Infrastructures im selben Bewerb
  - Berücksichtigung bei Punktevergabe und Teamgröße

Bewertung – lab2: Abgabeprotokoll (i)

- Analyse und Beschreibung der einzelnen Services
- Dokumentation der vorhandenen Schwachstellen bzw. Vermutungen
- Dokumentation der Lösungen / Lösungsansätze
- Beschreibung des Angriffswegs bzw. der Vermutungen
- Weitere (kreative) Ideen für Angriffe und Verteidigung
  - Auch anführen, wenn diese nicht durchgeführt wurden
  - Beispiel: automatisierte Angriffe
- ggf. Dokumentation zur Härtung des Systems

Bewertung – lab2: Abgabeprotokoll (ii)

- Anzahl der bearbeiteten/beschriebenen Services (Punkte pro Service)
- Automatisierung des Angriffs
- Automatisierung der Abgabe von Flags
- Punkteabzug bei formalen Fehlern (Vorlage vom Gameserver nicht verwendet, erforderliche Angaben nicht vorhanden, Namenskonventionen, Teammitglieder nicht angeführt, usw.)

WICHTIG: Es zählen auch Tätigkeiten, welche nicht zum gewünschten Erfolg führten. Z.B. fehlgeschlagene Angriffsversuche, gefundene Fehler ohne Behebung, usw.

Bewertung – Bewerb

- Verteidigung (Services müssen funktionieren, keine Angriffe)
- Angriff (Flags von anderen Teams sammeln)
- Laufende und funktionierende Services Voraussetzung für Angriffe
- Bonuspunkte
  - Abgabe von Advisories
  - Gute, kreative Lösungen (Beschreibung in Advisories)
  - Erhöhung der Gesamtpunkte um 15% bei erfolgreicher Abgabe min. eines Flags / Service
    (Visualisierung durch eine goldene Krone neben dem Team-Namen)
  - Teams mit meisten geknackten Services: silberne Krone

Punkte für Angriffe

- Nachweis eines erfolgreichen Angriffs: Abgabe eines gültigen Flags beim Gameserver
- Wiederholung: Flags sind für einen eingeschränkten Zeitraum gültig
- Verfügbarkeit des eigenen Services Voraussetzung zur erfolgreichen Flagabgabe
- Punkte bei einer erfolgreichen Abgabe eines gültigen Flags
  - Pro Service und Team volle Punkteanzahl für 4 Flags
  - Ab 5. Flag dieser Service-Team Kombination nur mehr 20% der Punkte $\rightarrow$ laufendes manuelles Ausnutzen des gleichen Services nicht zielführend
  - Keine Punkte für eigene Flags oder Test-Flags

Punkte für Verteidigung / Schadenspunkte

- Periodische Überprüfung der Services
- Punkte für „*erreichbare*“ Services
- *Weitere* Punkte für *funktionale* Services
- Status-Kennzeichnung am Gameserver
  - Vor Spielbeginn: **hidden**
  - Ab Spielbeginn: **up**, **down** oder **broken**
- Schadenspunkte bei erfolgreichem Angriff des eigenen Services in aktueller Runde
- Übersicht der Punkteverteilung am Gameserver

Auszug Allgemeine Regeln

- **Angriffe außerhalb der Übungsumgebung und auf die Infrastruktur für den Spielbetrieb sind verboten!**
- Konsequenzen vom Institut sowie der TU möglich
- Befolgung der Anweisungen des ESSE-Teams
- Regelverstöße können Punkteabzug bringen, gleich oder nach Analyse der Logs
- Kein „teamübergreifender“ Kontakt
- Kein Kontakt mit externen Personen
- Veröffentlichung der Detail-Regeln demnächst in TUWEL

Verbote bei Angriffen

- ARP-Spoofing
- DoS-Angriffe
- Flags löschen oder ändern
- Angriffe auf Notebooks
- **Angriffe außerhalb der Übungsumgebung und auf die Infrastruktur für den Spielbetrieb sind verboten!**

Verteidigung

- Verbote
  - Zugriffsbeschränkungen
    - Netzwerk-Ebene (z.B. Filterung auf Grund von IP-Adressen)
    - Applikations-Ebene (z.B. nur Gameserver erlauben)
  - Deaktivierung von spezifizierten Funktionen bei Services
- Gebote
  - Korrigieren Sie Schwachstellen; keine Work-Arounds!
  - Neukompilierung/Neustart von Services
  - Anpassung von Services/Scripts, solange die spezifizierte Funktion erhalten bleibt

Hinweise für die Vorgehensweise (i)

- Finden von Services
  - Benutzer am System (`/etc/passwd`, `/home/*`)
  - Untersuchung von laufenden Prozessen
  - Services in Docker Containern
  - Offene Ports und zugehörige Applikationen (`netstat`, `lsof`)
- Backups von relevanten Dateien vor Änderungen
  - *git* wird auf Team-Server vorab eingerichtet, nur Source-Files
  - Keine externen Kopien des kompletten Servers durchführen (Traffic!)

Hinweise für die Vorgehensweise (ii)

- Suche nach Sicherheitslücken und Korrektur dieser
- Log-Dateien überwachen; können wichtige Hinweise liefern, z.B.:
  - `tail -f datei`
  - `journalctl -f`
- ggf. Verwendung des eigenen Servers für Analysen
- Angriff der anderen Teams mittels gefundener Sicherheitslücken

Zusätzliche Hinweise / Tipps

- Protokoll nicht nur am Ende erstellen, sondern laufende Ergänzung
- Regelmäßige Überprüfung des Status Ihrer Services am Gameserver
  → Kennzeichnung von fehlerhaften Services
- Kontrolle von Nachrichten der Übungsleitung am Gameserver
- Server kann notfalls durch ESSE-Team zurückgesetzt werden
  → Verlust des bisherigen Setups
- Unterstützung durch das ESSE-Team bei Fragen
- Unmittelbare Meldung eventueller Regelverstöße / Verdacht auf unfaires Handeln an das ESSE-Team

Automatisierung

- Angriffe
- Abgabe von Flags
- Eigene Programme (JAVA, C++ etc.) möglich, wenn installiert
- Scripts
  - Bsp. für Scriptsprachen: Shell-Scripts (z.B. `bash`), `perl`, `python`
  - HTTP/HTTPS-Requests: `wget`, `curl`
  - Netzwerk: `telnet`, `netcat` (`nc`)
  - Filterung/Manipulation von Ergebnissen/Strings: `grep`, `sed`, `awk`
  - Netzwerk-Prozesse, offene Dateien/Ports: `netstat`, `lsof`
  - Dateien: `ls`, `find`, `cat`, `tac`, `tail`

Praktische Tipps zur Automatisierung

- Optimierung der automatischen Abgabe von Flags beim Gameserver
  - Abgabe nur der letzten Flags, nicht sämtliche gefundenen Flags
  - Timestamp zu Beginn der Flags (siehe Slide „Funktionsweise und Format von Flags“)
- Bei Scripts (unabsichtliches) Denial of Service verhindern, z.B. Sleep beim Abfragen
- TUWEL Ankündigungen demnächst
  - Interface-Beschreibung für Flag-Abgabe
  - Online Test-Service

Nachbesprechung

- Optional
- Anmeldung und Termin demnächst in TUWEL ersichtlich
- Ort wird spätestens nach Anmeldeende bekannt gegeben

Fragen

- Wenn Sie Fragen haben, stellen Sie sie jetzt!
- ... oder
  - Stellen Sie Ihre Frage im TUWEL-Übungsforum
  - Schreiben Sie uns ein e-mail: lva.security@inso.tuwien.ac.at

Weitere CTF-Bewerbe

- Teilnahme der ESSE bei internationalen CTF-Bewerben
- Bei bevorstehender Teilnahme Aussendung via Mailingliste
- Bei Interesse Informationen und Anmeldung zur Mailingliste auf
  - `http://www.defragmented-brains.at/`
- Meistens unterschiedlichste Kenntnisse erforderlich → keine Scheu vor Teilnahme

Vielen Dank!

https://security.inso.tuwien.ac.at/

Script-Beispiel – Abholung von Flags eines Web-Services

- *Hinweis*: Code dient der Illustration!

```
#! /bin/bash
rm flag
#getting flags
for i in `seq 100 1 108` `seq 110 1 111`; do
  sleep 1
  curl http://1.1.1.${i}/~chatservice/userlist | \
    sed 's/^.*::.*::.*::\(.*\)/\1/g' | \
    grep -v steve >> flag
done
```

Script-Beispiel – Abholung von Flags eines proprietären Services

- *Hinweis*: Code dient der Illustration!

```
#! /bin/bash
rm data
for i in `seq 100 1 108` `seq 110 1 111`; do
  sleep 1
  echo $i
  nc 1.1.1.${i} 3553 -w 1 < input | \
    grep patent-idea >> data
done
```