A process has been developed to transfer network intrusion data captured by Fail2ban to an adaptive enterprise intrusion detection and prevention system. The process involves software agents that we have created that are interconnected to a central behavior analysis database service where each software agent records attack meta-information collected during previous intrusion attempts. These distributed agents are the first phase of an overall plan to create a smarter network defense system through the collection and analysis of network signatures generated by real security threats. The central database to which the agents report warehouses and analyzes the meta-information collected by the interconnected agents. The agents can then utilize both instantaneous and historical data by integrating rules derived from the data collection and analysis process into intrusion prevention policies. The final result will be a modular and scalable network defense system that should be more responsive and adaptable to imminent threats.

Development Tools
- JetBrains PyCharm Professional IDE
  - Remote Development (Windows 10 ↔ Ubuntu master node)
  - Version Control (Windows 10 ↔ Remote Repo)
  - SSH/SFTP/HTTPS
- Emacs 25 - SLIME
- VMWare Workstation 12 PRO
  - Virtual Machines (10-50)
  - Memory: 4GB
  - 1 Processor 2 Cores (i7 5930k)
- Source Control
  - BitBucket – Private Repo: Master, Development Branches
  - GitHub – Private Repo: Redundant Repo Backup
  - Local Repo and Google Drive for – Document Repo
- Development Host Machine
  - i7 5930k (1 processor, 6 cores, 12 logical)
  - 32GB DDR4 RAM
  - 256GB SSD
- Development Languages
  - Python 2.7.X
  - Python comma separated value library
- Code Profilers
  - cProfile – Standard Library
  - cProfile – Standard Library
  - Line_profiler – External Library
- Code Libraries
  - External
    - Paramiko: Provides SSHv2 protocol
    - PyCrypto: Provides low-level cryptography
  - Internal
    - RE: Python/Perl regular expression library
    - CSV: Python comma separated value library
    - OS: Grants python access to underlying OS function calls
    - GLOB: Unix pathname pattern extension
    - CSVWriter.py

High-level diagram of the proposed adaptive enterprise intrusion detection and prevention system.

Process Environment and Workflow
A) Attack attempted on servers
B) Servers running Fail2ban
C) Jail log files of banned IP addresses
D) Process on collector server retrieves and parses the log files, and then writes data to csv files
E) Parsed csv files to be used as input for analysis

Troubleshooting Tools
- Code Profilers
  - cProfile – Standard Library
  - Line_profiler – External Library
- Code Visualization
  - Pycallgraph
  - PyCharm built-in callgraph
  - Snakefood python dependency graph

Code Samples
1.过程尝试将数据从CSV文件中抽取到Fail2ban的系统。
2.将数据从CSV文件中抽取到Fail2ban的系统。
3.将数据从CSV文件中抽取到Fail2ban的系统。
4.将数据从CSV文件中抽取到Fail2ban的系统。

Knowledge Discovery Process
1.收集数据
2.数据预处理
3.数据变换
4.数据挖掘
5.知识发现/评估

Risks
- Misidentification of benign activity as a threat (a false positive)
- System missing a threat, whereas doing nothing would allow the threat to pass
- Not a panacea for all network intrusion ill

Conclusions
The next step in our overall development of an adaptive enterprise intrusion detection and prevention system is to take the parsed output files, along with other source files, and start to construct the threat analysis clearinghouse. Another next step is to modify other network intrusion detection systems such as Snort to provide network intrusion data from a different angle. The modularity of this system provides flexibility while the uniqueness of this solution lies in the sum of its parts. Its modular, open-source design allows for lower costs, scalability, and for the system to be tailored to meet an organization’s needs. This will allow for the creation of an open alliance to change how threats are mitigated through a trusted and shared knowledge about attacks.