Pacific Northwest
NATIONAL LABORATORY

Flexible and Adaptive Malware Identification Using Techniques from Biology

March 21, 2022

Elena Peterson
Senior Cyber Security Researcher

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

Pacific Northwest
NATIONAL LABORATORY

DOE's 17 **national laboratories** tackle critical scientific challenges

Pacific Northwest NATIONAL LABORATORY

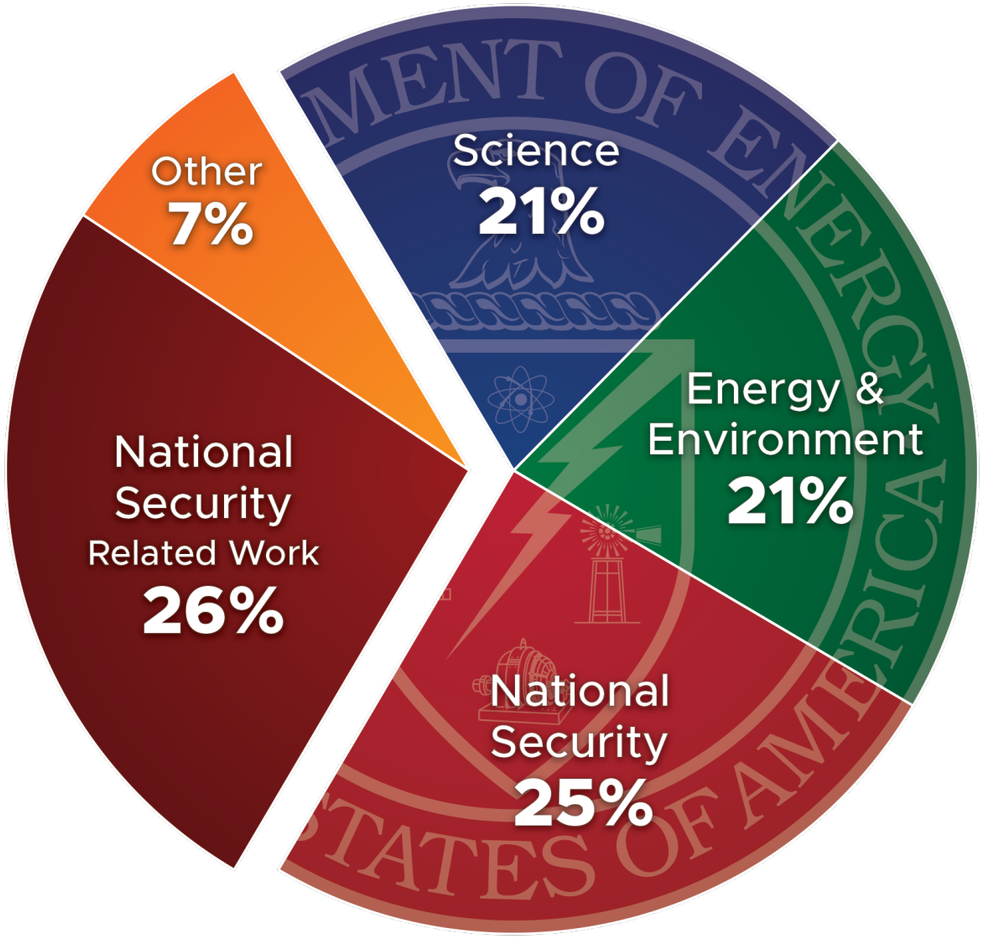
PNNL is DOE's **most diverse national laboratory**

$1.24B Spending

5,300 Staff

1,755 Peer-reviewed Publications

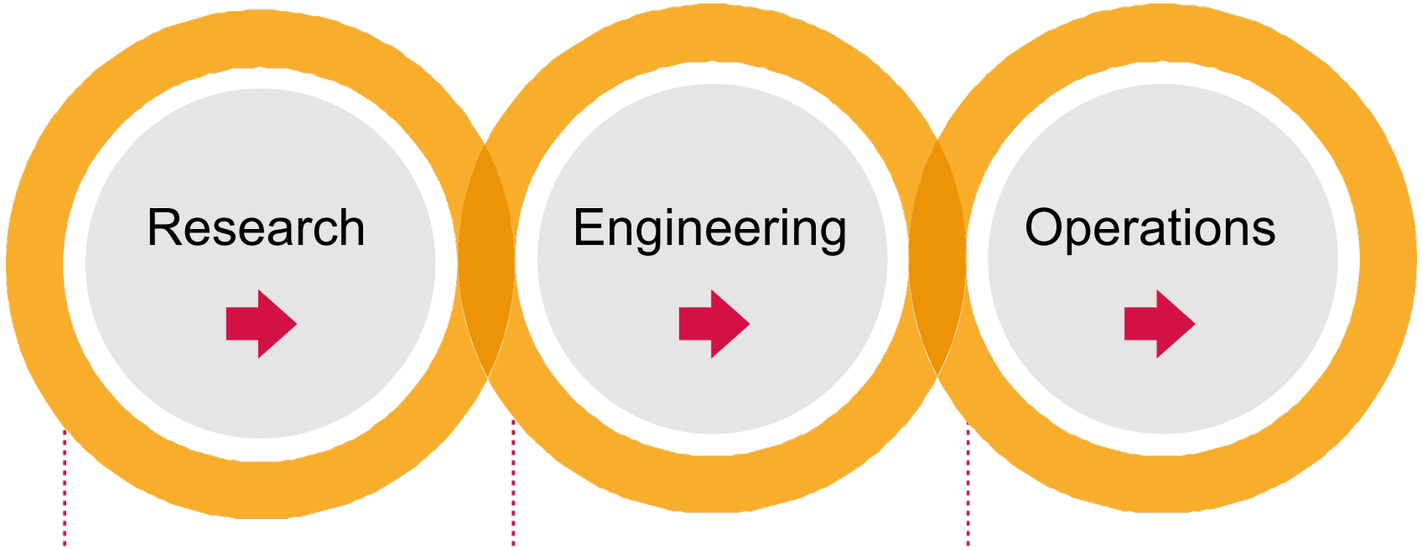
247 Invention Disclosures



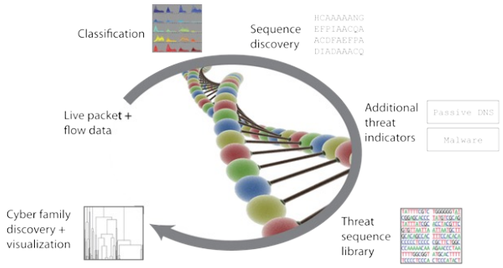
FY 2021 Spending

Pacific Northwest
NATIONAL LABORATORY

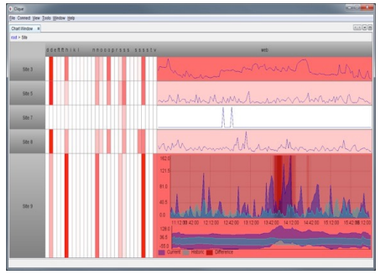
Mission-Driven Computing R&D



Scientists developed bio-inspired sequence alignment for cyber data



Engineers integrated anomaly models into a visualization platform



Analysts provide cyber threat intelligence to U.S. critical infrastructures

Pacific Northwest NATIONAL LABORATORY

MLSTONES Capability

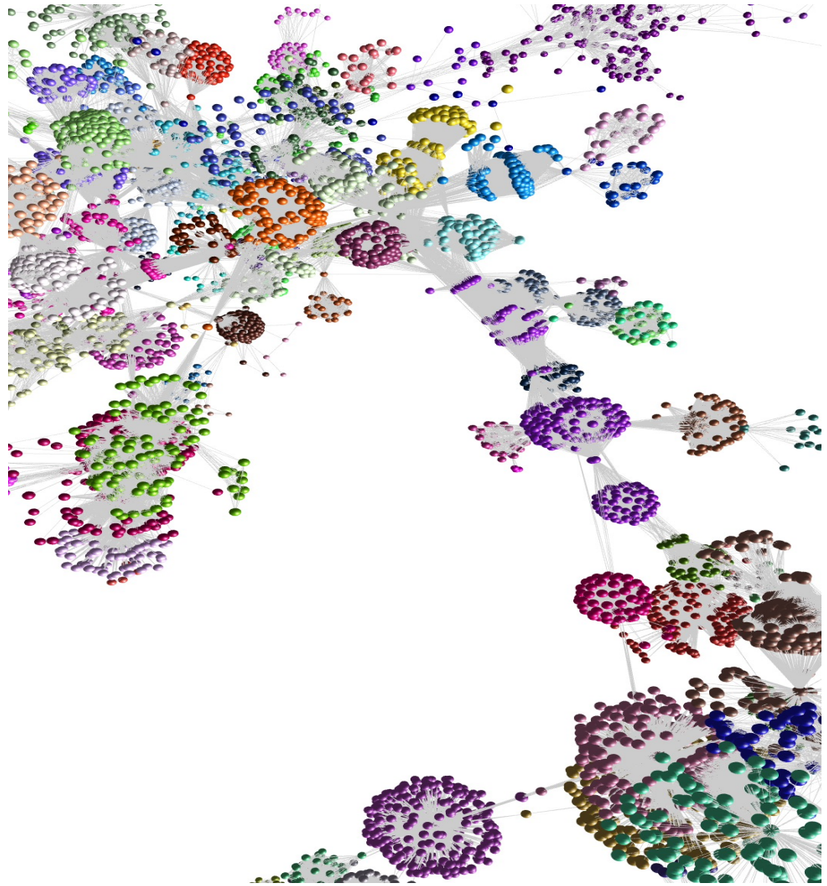
- Compare sets of *behavior*
  - Similarity score based on nearness metrics
- Behavior → ordered sequence of *events*
  - Order matters (frequency doesn't)
  - Time, sequence, domain specific ordering
- Events → anything that can be consistently labelled and grouped
  - Opcodes, log file entries, patterns of netflow, real-time events, …
- Leverages the nature of protein similarity
  - Proteins with similar structure have similar function
  - Determine function based on that similarity
  - Bioinformatics tools and algorithms

"Biosequence-based approach to analyzing binaries," issued January 29, 2019 (U.S. Patent No. 10,191,726)".

Pacific Northwest
NATIONAL LABORATORY

MLSTONES Use Cases

- Characterize behaviors
  - Cluster into families
  - Reduce data for faster comparisons
- Find new (similar) things
  - Similar but not exact
  - Need ground truth for identifying
- Identify new signals
  - Quickly
  - Don’t have to be looking for something specific
- Examples include (but not limited to):
  - Binary analysis (malware or not)
  - Changes in individual (IP) behavior based on netflow
  - Changes in individual behavior based on log files



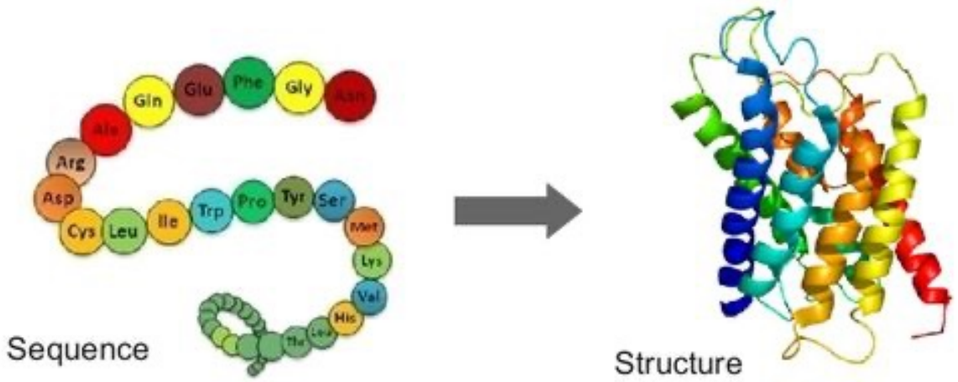
Malware use case

- Malware detection and classification is hard
- Easy and cheap to fool detection methods
  - Many methods rely on exact matching
    - ✓ MD5 hash to identify software
    - ✓ Rules to detect intrusion
  - Exact matching is easily defeated
- Must find an event by being hit with it before it can be detected
  - Can’t get MD5 hash without already identifying code as “of interest”
  - Can’t extract rules without experiencing exploit once
  - All variants of the same theme look like different events
- Algorithms must operate at data-intensive scales to be tenable

Now for the biology – protein similarity

- Information is encoded and passed down through genes to proteins
  - Proteins are the functional units performing most of the chemical and structural tasks needed for survival
  - The 20 amino acids can be mapped to 20 alpha characters and represented as text strings

AGHTVFDSQRSSVVPPAMMDFG

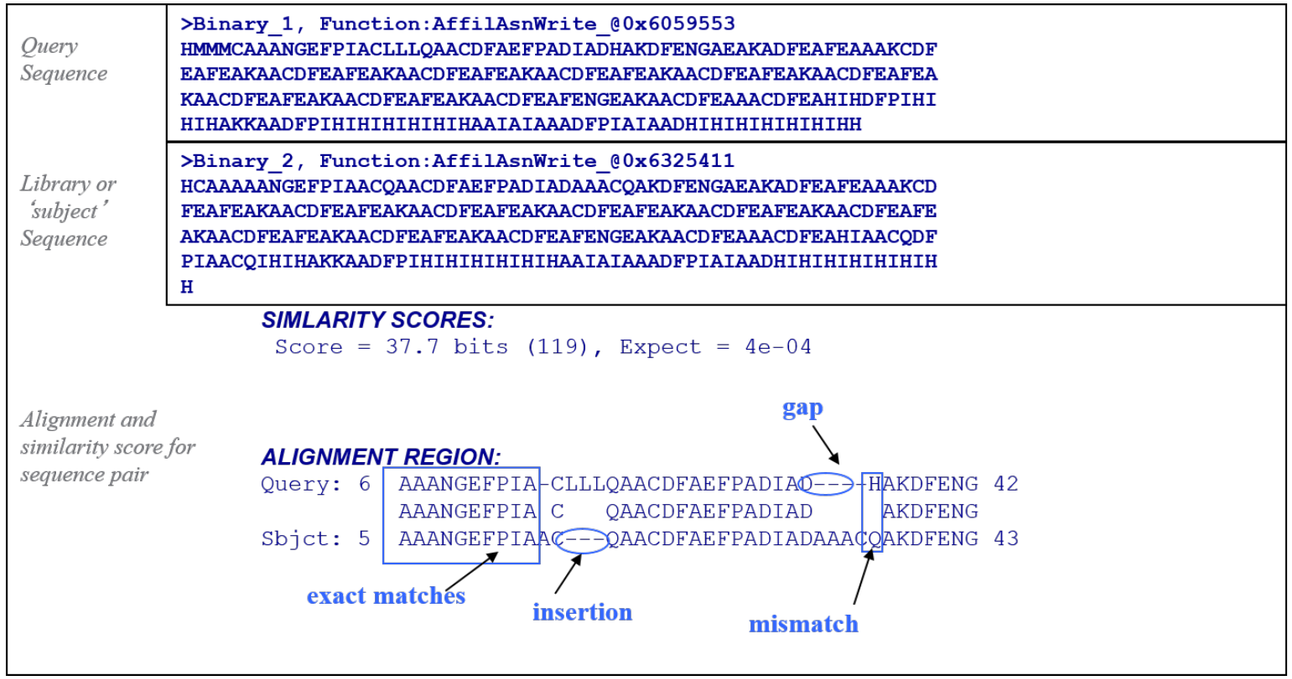


The sequence of a protein is highly related to its function:

- So when sequences are similar their function are very likely to be similar
- Matches don’t have to be “exact”

Binary (Malware) similarity

- Characterize and/or identify malware
  - Reduce data with motifs
  - Identify “new” malware through similarity
  - Other characterizations
    - ✓ Compilers
    - ✓ Compiler options
- Two-Step process:
  - Disassemble binary (YADD)
    - ✓ Functional blocks
    - ✓ Opcodes not operands
    - ✓ Bin opcodes
  - Compare using heuristics/stats (MADBlast)
    - ✓ Compare at functional level
    - ✓ Similar but not exact – variety of scores available
    - ✓ Need ground truth (similar to ML)



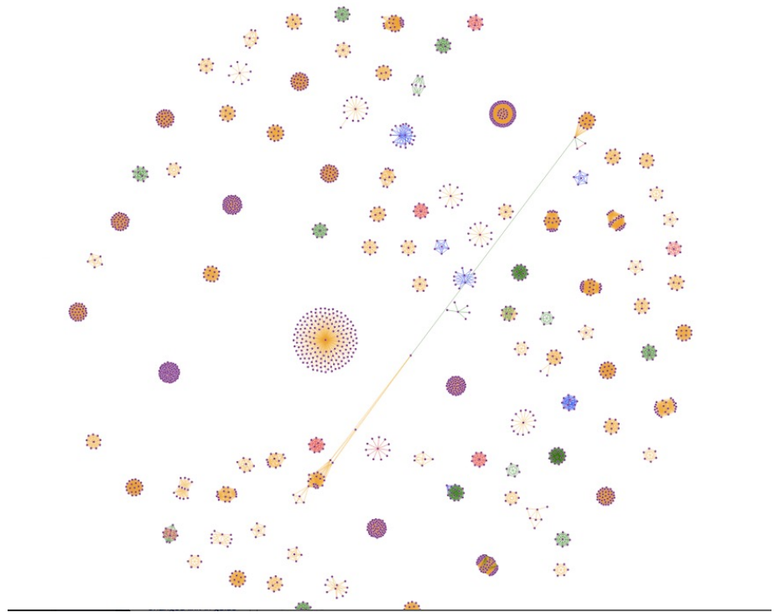
Mov → A
Push → B
Jmp → C

>Binary_1-Function_1
AABBCCFGMIDKRWOMAIAAB

Pacific Northwest
NATIONAL LABORATORY

Building Families

- Various clustering algorithms
- Based on MADBLAST output (scores of similarity)
- Definition of family parameters takes experimentation
  - Domain dependent
- Variations in homology are informative
- Could eventually be used in phylogeny
  - Understanding the family tree
  - Evolution



Motif Building with Consensus and Profiles

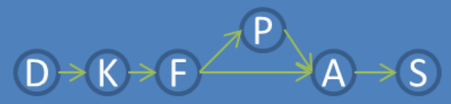
Family Alignment

```
1 ASLDKFPASRDASLPFSAEF
2 ASLDKF-ASRDASL-FSAEF
3 ASLDKF-ASRDASL-FSAEF
4 ASLDKF-ASDDASL--SAEF
5 ASLDKF-ASDDASL--SAEF
  ****** ** ****  ****
```

Consensus sequence signature

ASLDKFXASXDASLXXSAEF

Profile-based signature



Pacific Northwest NATIONAL LABORATORY

Family Example

```
mov     ebx, 0A34ED533h
xor     edi, edi

loc_637BBC:
cmp     edi, 6B4h
jz      short loc_637BE0
sub     edi, 0FFFFFFFCh
xor     esi, esi
push    dword ptr [ecx+esi]

sub     [esp], ebx


pop     dword ptr [edx+esi*2]

dec     dword ptr [edx]
xor     ebx, ebx

add     ebx, [ecx]
add     ecx, 4
add     edx, 4
jmp     short loc_637BBC
loc_637BE0:
```

```
mov     ecx, 987AD624h ; (crypto key)
xor     ebx, ebx

loc_635A08:
cmp     ebx, 6ACh ; (memory size)
jz      short loc_635A2E
sub     ebx, 0FFFFFFFCh
xor     edi, edi
push    dword ptr [esi+edi]

inc     [esp] ; (de/encryption method)
xor     [esp], ecx

pop     dword ptr [edx+edi*2]

sub     ecx, ecx
dec     ecx

and     ecx, [esi]
add     esi, 4
add     edx, 4
jmp     short loc_635A08
loc_635A2E:
```

Pacific Northwest NATIONAL LABORATORY

Family Example

- Similar but non-identical sequences are grouped together in a family
- Differences are non-trivial
  - Different registers
  - Different encryption keys
  - Different memory-region sizes
  - Different encryption/decryption operation
- Putting all of this annotation into infrastructure for query and sharing
  - **Functions are grouped into families based on similar structure (not exact matching)**
  - **These families represent similar behaviors**
  - **The concept of families (clustering based on structure) representing behavior holds true**

Pacific Northwest
NATIONAL LABORATORY

Compiler Differences – 2 key questions

- Do differences in compiler options limit our ability to detect similar binaries using MLSTONES?

- Are similar binaries still detectable when the same source code is built using different compilers or compiler versions?

Campaign 1 Result

- O0 very different
  - Lack of optimization
  - Assembly is as close to the source code as possible
- Og and O1 group together
  - Og does perform optimization (weak)
  - O1 is a weak version of optimization
  - Uses similar flags
- O2, O3, Ofast group together
  - O2 aggressive
  - O3 and Ofast similarly aggressive
  - Uses similar flags

Pacific Northwest NATIONAL LABORATORY

Campaign 2 results using NMAP

	clang_3.4	clang_3.5	clang_3.6	gcc_4.7	gcc_4.8	gcc_4.9	gcc_5	intel_2016	intel_2013	intel_2015
clang_3.4	1	0.994652406	0.973262032	0.021390374	0.021390374	0.026737968	0.026737968	0.026737968	0.026737968	0.026737968
clang_3.5	1	1	0.978378378	0.016216216	0.016216216	0.027027027	0.027027027	0.027027027	0.027027027	0.027027027
clang_3.6	0.982758621	0.988505747	1	0.017241379	0.017241379	0.028735632	0.028735632	0.028735632	0.028735632	0.028735632
gcc_4.7	0.009259259	0.009259259	0.009259259	1	0.689814815	0.101851852	0.101851852	0.009259259	0.009259259	0.009259259
gcc_4.8	0.011627907	0.011627907	0.011627907	0.843023256	1	0.226744186	0.13372093	0.011627907	0.011627907	0.011627907
gcc_4.9	0.003333333	0.003333333	0.003333333	0.010666667	0.022666667	1	0.763333333	0.003333333	0.003333333	0.003333333
gcc_5	0.003194888	0.003194888	0.003194888	0.015335463	0.016613419	0.715015974	1	0.003194888	0.003194888	0.003194888
intel_2016	0.014204545	0.014204545	0.014204545	0.005681818	0.005681818	0.011363636	0.011363636	1	0.886363636	0.909090909
intel_2013	0.013812155	0.013812155	0.013812155	0.005524862	0.005524862	0.011049724	0.011049724	0.809392265	1	0.986187845
intel_2015	0.013927577	0.013927577	0.013927577	0.005571031	0.005571031	0.011142061	0.011142061	0.824512535	0.977715877	1

Pacific Northwest NATIONAL LABORATORY

Compiler Results

- Our process can differentiate between compilers
  - And sometimes versions of compilers
- Our process can differentiate between groups of compiler options

- This information can be used in a variety of ways
  - Attribution
  - Searching or dividing up a catalog of malware
  - Support for reverse engineering
  - Prediction of new variants (to be tested)

Pacific Northwest
NATIONAL LABORATORY

Its not perfect

- Malware
  - Some obfuscation techniques are not mitigated natively
    - ✓ Packers
    - ✓ DLL Injection
  - Fuzzy/Lossy process
    - ✓ Can’t be reverse engineered which means housekeeping is key
    - ✓ Understand FP/FN/TP/TN is a must
- Other domains
  - Mappings created (almost) by hand
  - Testing/experiments are required to assess effectiveness
  - Doesn’t work well with small data

Pacific Northwest
NATIONAL LABORATORY

Binary Diversity (Dissimilarity)

- Same general process (disassemble and compare)
- Different “scoring” methodology
- No need for ground truth
- Small datasets (relatively)
- X86, ARM, MIPS, POWER, and RISC-V
- Tested on good code and malware
  - Compared favorable to other tools
- Capability is available
  - Standalone VM or Docker container
  - API ready – easily integrated

The project team

- Elena Peterson – Software/Cyber Engineer
- Chris Oehmen – HPC/Biology
- Aaron Phillips – Software Engineer
- Richard Griswold – Reverse Engineer
- Brett Jefferson - Math and Statistics
- Blaine McGarry – Lead UI Developer
- Tyler Williams – Malware Expert
- Taylor Edwards, Keith Star – software development
- Cameron Smith – Reverse Engineer**
- Arne Nixon - intern

Peterson, Elena S., et al. "Flexible and Adaptive Malware Identification Using Techniques from Biology." *Journal of Information Warfare* 20.PNNL-SA-154741 (2021).

Pacific Northwest
NATIONAL LABORATORY

Partnering and Collaboration

Government

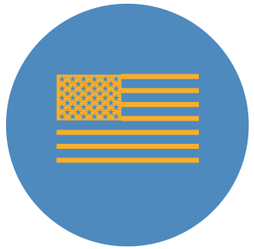
- Staff exchanges and “analysts in residence;” Workforce 21 externships
- Technology insertion assistance and lessons learned assessment

Academia

- National Security Internship Program
- Faculty Summits, Faculty in residence, joint university appointments

Industry

- Seattle Innovation District engagements
- Testbeds, technology assessment



Workforce Development

- Externships for practicing analysts
- Software Carpentry, data science courses

Pacific Northwest NATIONAL LABORATORY

Working with Us

- **Joint research and development**
  - Access to PNNL testbeds and test ranges, and real-world data
  - PNNL funding on internal R&D, especially for grad students
  - Joint proposal development for sponsored programs
  - Visiting Faculty Program
- **Internships for undergrads and grad students**
  - National Security Internship Program includes eligibility for tuition reimbursement
- **Staff positions for computing and cyber researchers, cyber defenders**
  - Including joint appointments with universities

Learn more

- http://pnnl.gov/computing
- http://workbasedlearning.pnnl.gov/
- http://jobs.pnnl.gov

Pacific Northwest
NATIONAL LABORATORY

Elena Peterson
Senior Cyber Security Researcher

Elena@pnnl.gov

National Security Directorate

cybersecurity.pnnl.gov

Pacific Northwest
NATIONAL LABORATORY

Thank you