

**Update to the 2016 Federal Cybersecurity Research and Development
Strategic Plan RFI Responses**

DISCLAIMER: [The RFI public responses](#) received and posted do not represent the views and/or opinions of the U.S. Government, NSTC Subcommittee on Networking and Information Technology Research and Development (NITRD), NITRD National Coordination Office, and/or any other Federal agencies and/or government entities. We bear no responsibility for the accuracy, legality or content of all external links included in this document.

From: David Martinez

Subject: RFI Response: Federal Cybersecurity R&D Strategic Plan

5. What changes to cybersecurity education and workforce development, at all levels of education, should be considered to prepare students, faculty, and the workforce in the next decade for emerging cybersecurity challenges, such as the implications of artificial intelligence, quantum computing, and the Internet of Things on cybersecurity?

- Computer History: Given the pervasiveness of technology in modern society, foundational computer history should be taught in K-12 schools. In some instances computer history may be weaved into existing history curriculums by highlighting significant computer related events that intersect or overlap historical events and milestones currently covered in K-12 schools (ie. A protracted detailing of the significance of cryptography during WW2 and the role that Alan Turing and others played in breaking the Enigma cypher.) In other instances a history curriculum highlighting the pioneers of computing, telecommunications, and other technologies may be appropriate. For students, an understanding of the conception of computing through modern day ubiquity will help illustrate the underlying technologies and sciences that create the modern systems students interact with daily. The historical curriculum may also help reveal the various specializations that one may pursue to support the development of new technologies. [NOTE: 'The Innovators' by Walter Isaacson provides a good examples of development of modern computing over the last two centuries. 'Code: The Hidden Language of Computer Hardware and Software' by Charles Petzold is another good, albeit more technical example of the development of modern computing]

- Personal Security in an Online World (ie. Social Engineering, IT security best practices) : Most individuals will begin interacting with technology long before being given any significant guidance on security best practices. The annual list of most common passwords continues to contain many of the same insecure passwords year-over-year. The foundations of good computer security should be taught in early elementary education as young students are just beginning to interact with online technologies. Students should also be trained on social engineering tactics and how to protect oneself. Examples of social engineering exist throughout history, though in the era of modern computing social engineering has evolved into a multifaceted threat that could impact every citizen. Understanding social engineering, the current threat landscape, and mitigations, is vital for protecting individuals, corporations, and government. In most cases it seems that the average individual doesn't receive training on combating social engineering until they've joined an institution much later in life, if at all, and even then the training is oftentimes rudimentary. By the time most receive this training they've had years of leaking immutable personal information such as familial relationships and personally identifiable information. Especially as youth are independently accessing the internet, participating in social forums and online gaming, it's critical that they be educated on how to safely utilize technologies and protect their information and sensitive data. Social engineering remains the leading technique utilized by cyber criminals and nation states to compromise system and individuals.

- Basics of computer programming: With the expanding need (and shortage) of programmers, students should be introduced to the fundamentals of programming. Although there are several languages, some which vary greatly, there are some fundamentals that could be introduced to familiarize students with basic programming concepts.

- Fundamental Windows, MS Office, and other software training: I often witness a tremendous amount of lost productivity as a result of workers being ignorant of the robust features of common software programs. Windows and Microsoft Office programs are heavily used in academia, government, business, and by private individuals. Yet, to my knowledge, there is little formal training provided through our education system and few individuals independently seek such training. A formal course introducing students to the features of these tools would likely increase productivity in education and beyond. This could be accomplished by offering dedicated courses, or by including instruction and practice of these tools within existing courses; teaching tools and techniques associated with each course/topic/assignment as appropriate. For example, a writing course could focus on features associated with Microsoft Word, and a mathematics course could highlight Excel functions.

Lastly, and more generally, introducing relational databases/data mining and critical thinking are also vital to the future success of these students. Nearly every field is now leveraging data warehouses and relational databases in some form. As AI/ML continue to advance, the focus on data, forecasting, and correlative efforts will further increase. Introducing students to these concepts and methodology will become increasingly important, particularly as AI and automation continue to eliminate more menial and repetitive tasks that are currently completed by a material portion of the US workforce.