Report on Patient Privacy Volume 20, Number 9. September 10, 2020 Privacy Briefs: September 2020

By Jane Anderson

◆ **Utah Pathology Services, based in Salt Lake City, has reported a data breach involving approximately 112,000 patients.** According to the medical practice's "Notice of Data Incident," the practice learned June 30 that "an unknown third party attempted to redirect funds from Utah Pathology."[1] The practice said that this suspicious activity "did not involve any patient information, or the completion of any financial transactions." Upon discovery of the attempted fraud, Utah Pathology said that it quickly secured the affected email account and launched an investigation, with assistance from independent information technology security and forensic investigators. "We discovered that the personal information of certain individuals, including names and one or more of the following personal attributes was accessible to the unauthorized party: date of birth, gender, phone number, mailing address, email address, insurance information including id and group numbers, medical and health information including: internal record numbers and clinical and diagnostic information related to pathology services, and, for a small percentage of patients, Social Security number." There's no evidence that the information has been misused, Utah Pathology said. The practice said it is implementing additional safeguards and security measures, and will provide identity monitoring to affected individuals for one year.

◆ **Developer error caused the leak of 150,000 to 200,000 patient health records stored in productivity apps from Microsoft and Google that were found on the site GitHub, according to a report.**[2] Dutch researcher Jelle Ursem discovered nine separate files of what was termed "highly sensitive" protected health information from nine separate health organizations. The apps involved included Microsoft's Office 365 and Google's G Suite. Ursem said he had difficulty reaching the companies whose data had been leaked, and therefore eventually reported the breach to DataBreaches.net, which worked with him to publish a collaborative paper on the findings. According to the paper, the information was exposed because of developers' improper configuration of access controls and hardcoded credentials in the storing of the information. The leaks were commonly caused by developers embedding hard-coded login credentials into code instead of making them a configuration option on the server, using public repositories instead of private repositories, failing to use two-factor or multifactor authentication for email accounts, and/or abandoning repositories instead of deleting them when no longer needed. In addition, the paper said, errors often went undetected for years because organizations failed to audit their developer's security and compliance with security policies, failed to have a monitored account for researchers to report security concerns, and failed to respond to attempts at responsible disclosure for fear that the notification was a social engineering hack.